

## Trabajo Fin de Máster

Diseño de un sistema de telegestión de  
contadores de gas domésticos y su integración en  
un sistema de telegestión de contadores eléctricos  
(embedded software)

Autor

Alejandro Cuadra Giménez

Director

Alfredo Sanz Molina

MÁSTER EN INGENIERÍA ELECTRÓNICA  
Escuela de Ingeniería y Arquitectura  
Área de Tecnología Electrónica  
Departamento de Ingeniería Electrónica y Comunicaciones

Zaragoza, junio de 2013



## **Diseño de un sistema de telegestión de contadores de gas domésticos y su integración en un sistema de telegestión de contadores eléctricos (embedded software)**

### Resumen

Debido al despliegue de los nuevos sistemas de telegestión de electricidad y dado que las compañías energéticas pretenden unificar los mercados de la electricidad y el gas, surge la necesidad de tener un sistema de telegestión de gas. Para ello aprovechando parte de la infraestructura ya desplegada del sistema de telegestión de electricidad se va a diseñar un sistema piloto de telegestión de los contadores de gas domésticos. Para conseguir este objetivo es necesario diseñar un dispositivo llamado Meter Interface (MI) que registre los datos del contador de gas y lo transmita a través de radiofrecuencia (RF) a otro dispositivo llamado Multi Utility Communication Controller (MUC) el cual debe recibir los datos de varios contadores de gas y transmitirlos por medio de Power Line Communication (PLC) a través de la red eléctrica al gestor del centro de transformación. De esta manera tendríamos un sólo sistema tanto para la gestión de los contadores de gas como para la de los contadores de electricidad.

El trabajo se enmarca en un proyecto de metering para gas encargado por una importante compañía del sector energético a la empresa Atmel. Las bases iniciales del sistema en general y de cada uno de los dispositivos en particular vienen dados por la compañía energética.

Para el MI algunos de los requisitos básicos son:

- Alimentación por baterías con un mínimo de autonomía.
- Detección correcta de la medición del consumo en el contador de gas.
- Implementación de determinadas alarmas.
- Registro de datos.
- Comunicación RF mediante un protocolo por desarrollar con un mínimo de cobertura.

Para el MUC algunos de los requisitos básicos son:

- Gestión de una red de varios MI.
- Implementación de determinadas alarmas.
- Registro de datos.
- Comunicación RF mediante un protocolo por desarrollar con un mínimo de cobertura.
- Comunicación local por puerto serie, mediante un protocolo por desarrollar.
- Comunicación PLC, mediante estándar de comunicación ya existente.

Por lo tanto el desarrollo del sistema y los equipos que lo conforman implica la definición de protocolos de comunicación como nuevos estándares, diseño hardware, elección de la envolvente, diseño firmware (embedded software) y testing.

Debido a la elevada carga de trabajo que conlleva el proyecto, en Atmel se decidió dividirlo en dos partes, una centrada en el hardware y otra en el firmware, quedando el resto de tareas como compartidas.

Este trabajo en concreto recoge la parte centrada en embedded software.



**ÍNDICE**

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>11</b>
1.1	Situación .....	11
1.2	Objetivos y alcance .....	11
1.3	Requisitos .....	11
1.3.1	Descripción general del sistema.....	12
1.3.2	Protocolos de comunicación .....	13
1.3.3	Funcionalidades.....	14
1.4	Estado del Arte .....	15
1.5	Planificación .....	17
<b>2</b>	<b>PROTOCOLOS DE COMUNICACIÓN .....</b>	<b>19</b>
2.1	Radiofrecuencia.....	19
2.2	Power Line Communications .....	20
2.3	Puerto serie.....	20
<b>3</b>	<b>FIRMWARE .....</b>	<b>21</b>
3.1	Local Data Concentrator .....	21
3.1.1	Esquema general .....	22
3.1.2	Bootloader .....	23
3.1.3	Application .....	23
3.1.3.1	Sistema en tiempo real .....	23
3.1.3.2	Módulo Startup .....	23
3.1.3.3	Módulo COSEM.....	24
3.1.3.4	Módulo PLC .....	25
3.1.3.5	Módulo Image Transfer .....	25
3.1.3.6	Módulo BaseApp .....	26
3.1.3.6.1	Submódulo WM-Bus .....	26
3.1.3.6.2	Submódulo Backdoor.....	26
3.1.3.6.3	Submódulo Logs/Storage .....	26

3.1.3.6.4	Submódulo Driver de la flash externa.....	27
3.1.3.6.5	Submódulo FW Upgrade .....	27
3.1.3.6.6	Submódulo Debug Traces .....	27
3.1.3.6.7	Gestión de tareas.....	27
<b>3.2</b>	<b>Endpoint.....</b>	<b>28</b>
3.2.1	Esquema general .....	29
3.2.2	Bootloader.....	30
3.2.3	Application .....	30
3.2.3.1	Sistema en tiempo real.....	30
3.2.3.2	Módulo Startup .....	31
3.2.3.3	Módulo WMBus.....	31
3.2.3.4	Módulo Logs/Storage .....	32
3.2.3.5	Módulo FW Upgrade .....	32
3.2.3.6	Módulo extEEPROM .....	32
3.2.3.7	Módulo BaseApp .....	33
3.2.3.7.1	Submódulo Setup .....	33
3.2.3.7.2	Submódulo Alarms.....	33
3.2.3.7.3	Submódulo Report.....	33
<b>4</b>	<b>TESTING .....</b>	<b>35</b>
<b>5</b>	<b>CONCLUSIONES.....</b>	<b>37</b>
5.1	Consecución de Objetivos.....	37
5.2	Líneas de Trabajo Futuras .....	38
<b>6</b>	<b>REFERENCIAS .....</b>	<b>39</b>
<b>7</b>	<b>GLOSARIO.....</b>	<b>41</b>
<b>ANEXO A</b>	<b>PLANIFICACIÓN .....</b>	<b>43</b>
A.1	Fases y tareas.....	43
A.2	Cronograma .....	44
<b>ANEXO B</b>	<b>HARDWARE.....</b>	<b>47</b>
B.1	Local Data Concentrator .....	47

B.1.1	Esquema de placa.....	47
B.1.2	Compatibilidad .....	48
B.2	Endpoint.....	48
B.2.1	Esquema de placa.....	48
B.2.2	Compatibilidad .....	48
ANEXO C	HERRAMIENTAS DE DESARROLLO .....	49
C.1	Firmware .....	49
C.1.1	Local Data Concentrator.....	49
C.1.1.1	IAR.....	49
C.1.1.2	Programador J-Link .....	50
C.1.1.3	J-Link Commander .....	51
C.1.2	Endpoint .....	51
C.1.2.1	IAR.....	51
C.1.2.2	Programador JTAGICE mkII .....	51
C.1.2.3	Atmel Studio .....	51
C.2	Configuración y pruebas .....	52
C.2.1	Hand Held Terminal .....	52
C.2.1.1	RFI .....	52
C.2.1.2	Metering Field Tool 4.0.0.....	53
C.2.2	Sniffer WM-Bus .....	53
C.2.3	Aplicaciones Atmel para PLC y PRIME .....	54
C.2.4	Phenix.....	55
C.2.5	Gestor de Centro de Transformación.....	56
C.2.6	Imán .....	56
C.2.7	Compresor de aire .....	56
C.2.8	Fuentes de alimentación .....	56
C.2.9	Contadores de gas.....	57
C.2.10	Ordenadores personales .....	57





## **ÍNDICE DE FIGURAS**

Figura 1	
Esquema del sistema.....	12
Figura 2	
EP (izda.) y LDC (dcha.) .....	14
Figura 3	
Contador de gas marca Itron (izda.) y EP montado en contador de gas (dcha.) .....	14
Figura 4	
Esquema FW del LDC.....	22
Figura 5	
Esquema FW del EP.....	29
Figura 6	
Cronograma. ....	45
Figura 7	
Esquema HW del LDC. ....	47
Figura 8	
Esquema HW del EP.....	48
Figura 9	
IDE de IAR Embedded Workbench.....	49
Figura 10	
Programador J-Link.....	50
Figura 11	
Entorno JLink-Commander. ....	50
Figura 12	
Programador JTAGICE mkII.....	51
Figura 13	
Atmel Studio 6. ....	52
Figura 14	
RFI.....	53
Figura 15	
Interfaz de inicio de la MFT.....	53
Figura 16	
Sniffer WM-Bus iM871A IMST. ....	54
Figura 17	
Wireless M-Bus Studio 0.1.9.....	54
Figura 18	
ADD6030v30 v1.6.09. ....	55
Figura 19	
Vista de los test suites de Phenix. ....	55

Figura 20	
Interfaz web de CCT4. ....	56
Figura 21	
Contadores de gas marcas Itron (izda.) y Elster (dcha.). ....	57

## **1 INTRODUCCIÓN**

### **1.1 Situación**

El presente trabajo ha sido realizado en el seno de la empresa Atmel Corporation [1], multinacional americana de componentes y productos electrónicos, concretamente en su filial española Atmel Spain, ubicada en Zaragoza.

Atmel Spain desarrolla productos de Power Line Communications (PLC) en aplicaciones relacionadas con la gestión de la energía y proporciona soporte al cliente dentro de este campo en todo el mundo.

Tras recibir una invitación de una importante compañía energética española, para formar parte de un consorcio para el desarrollo de nuevos estándares y equipos destinados a servicios de telegestión de gas, Atmel Spain crea un proyecto I+D+i de gran envergadura, enmarcado en la denominación Automatic Meter Management (AMM).

### **1.2 Objetivos y alcance**

El proyecto completo tiene como objetivo el diseño de un sistema piloto de telegestión de contadores de gas domésticos, aprovechando parte de la infraestructura del sistema de telegestión de electricidad existente, de manera que ambos sistemas queden integrados formando parte de un único sistema global de telemetría. Además, se requiere la interoperabilidad con los equipos desarrollados por otras empresas internacionales, que también se encuentran dentro del consorcio.

La integración propuesta se describe en posteriores apartados, aunque conviene adelantar que el desarrollo del sistema y los equipos que lo conforman implica la definición de protocolos de comunicación como nuevos estándares, diseño hardware, elección de envoltentes, diseño firmware (embedded software) y testing.

Dada la magnitud del proyecto, en Atmel se decidió dividirlo en dos partes, una centrada en el hardware y otra en el firmware, quedando el resto de tareas como compartidas.

El objetivo particular de este trabajo es la colaboración en la definición de los nuevos protocolos de comunicación dentro del consorcio, así como el desarrollo de la estructura del firmware y parte de su programación en determinados microcontroladores (MCU) necesarios para gestionar los dispositivos. Además, incluye la supervisión general del firmware y el completo testeo del mismo, tanto de programación propia como realizado por otros miembros del equipo Atmel. No obstante, la presente memoria se centra en la descripción de la estructura del firmware, sin profundizar demasiado en los otros aspectos, debido a las restricciones impuestas tanto por Atmel como por la compañía energética.

### **1.3 Requisitos**

En los inicios del proyecto, el consorcio (Atmel incluida) definió las especificaciones básicas como criterios a seguir, fase en la que el autor de este trabajo todavía no formaba parte del

proyecto. En adelante se hará referencia a estas especificaciones como especificaciones del consorcio.

En dichas especificaciones, ya se definían los requisitos básicos a cumplir por el sistema, como la configuración de red, los protocolos de comunicación a utilizar, los elementos que lo componen y las funcionalidades de los mismos. Consecuentemente, la labor desempeñada por el autor, en el marco de sus objetivos dentro del proyecto, ha consistido en dar forma a todos estos puntos, siguiendo las directrices marcadas, para completar y finalizar las definiciones iniciales.

Estas especificaciones no son de dominio público, por lo que en este trabajo sólo se describen sucintamente.

### 1.3.1 Descripción general del sistema

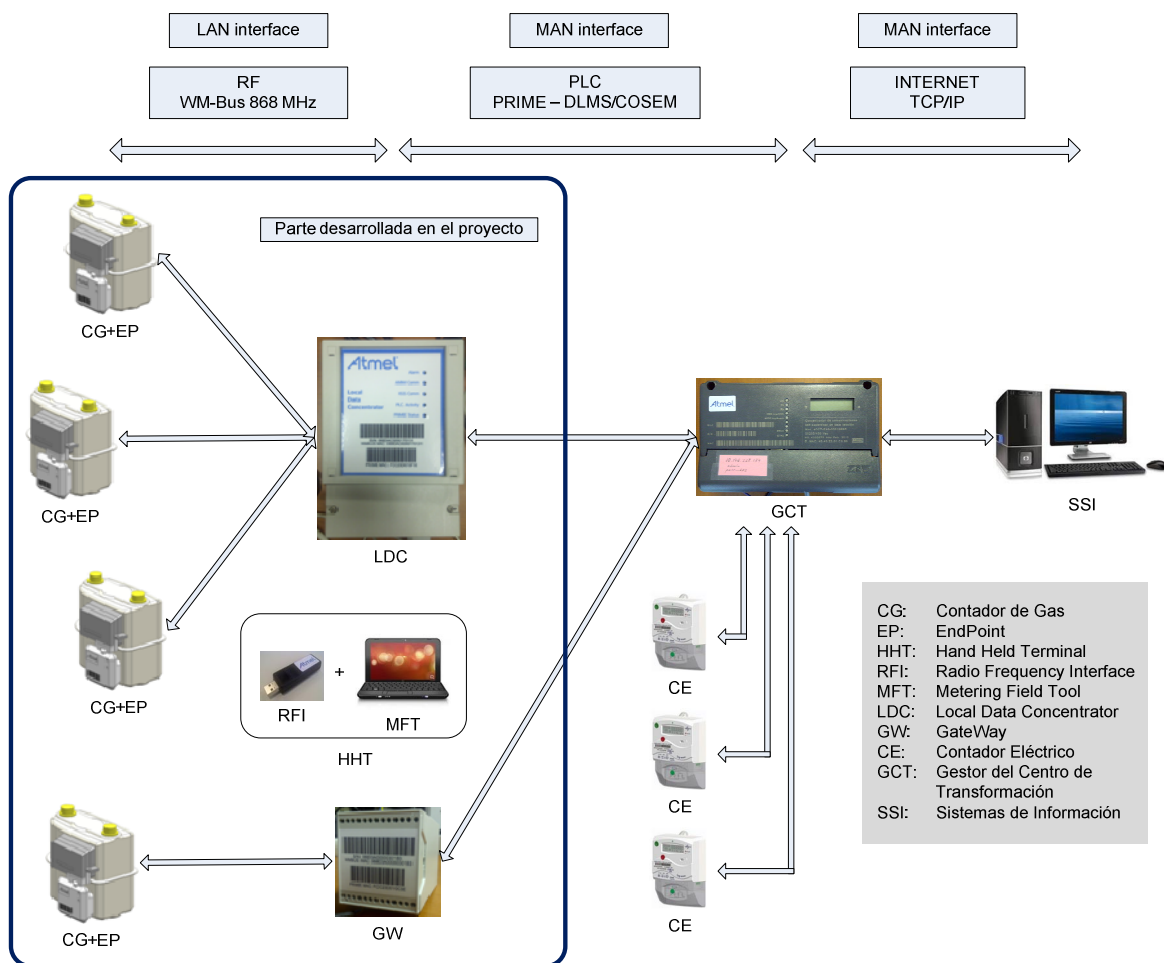


Figura 1. Esquema del sistema.

El sistema de telegestión de gas debe estar formado por un dispositivo llamado inicialmente Meter Interface (MI) que registre los datos del contador de gas y los transmita a través de radiofrecuencia (RF), ya que los recintos de contadores de gas no permiten líneas eléctricas (atmósfera ATEX), a otro dispositivo llamado inicialmente Multi Utility Communication Controller (MUC), el cual debe recibir los datos de varios contadores de gas y transmitirlos por medio de Power Line Communications (PLC) a través de la red eléctrica al Gestor del Centro de Transformación (GCT). Este último estará conectado vía internet con los Sistemas de Información

(SSI) de la compañía energética para permitir la gestión remota. Para la configuración y gestión tanto del MI como del MUC se requiere de un equipo llamado Hand Held Terminal (HHT).

Las denominaciones genéricas MI y MUC, usadas en las especificaciones, fueron modificadas en el ámbito del proyecto, renombrándose el MI como Endpoint (EP) y dividiéndose el MUC en dos equipos distintos, llamados Local Data Concentrator (LDC) y Gateway (GW). En adelante se utilizará esta nueva nomenclatura.

En la Figura 1 se muestra el esquema completo del sistema, aunque la parte objeto de desarrollo sólo abarca desde el EP hasta el LDC/GW.

A continuación se describen los elementos del sistema:

- **Endpoint**: elemento final de la red, responsable del proceso de lectura. Es el encargado de adquirir, procesar y enviar las lecturas del contador de gas, así como el estado de las alarmas y los logs, al concentrador (LDC o GW).
- **Local Data Concentrator**: dispositivo que recibe la información (lecturas, alarmas y logs) de un grupo de EP a través de RF y la transmite por PLC al GCT. A su vez, actúa como elemento transmisor de las instrucciones de gestión hacia los EP recibidas del GCT.
- **Gateway**: funcionalmente idéntico al LDC, pero limitado a la gestión de un solo EP.
- **Gestor del Centro de Transformación**: elemento del sistema, ya existente en redes de metrología para electricidad, capaz de concentrar las comunicaciones procedentes de los LDC/GW (así como de contadores eléctricos) y transmitirlos por TCP/IP a los SSI.
- **Sistemas de Información**: presentes en las instalaciones de la compañía energética, los sistemas de información almacenan los datos y aplicaciones del sistema AMM, posibilitando su gestión remota. Conectados a los GCT por TCP/IP.
- **Hand Held Terminal**: equipo portátil para las tareas de configuración y gestión, tanto del EP como del LDC/GW. Está formado por un dispositivo RF llamado Radio Frequency Interface (RFI), conectado a un PC por USB para ser gestionado por una aplicación llamada Metering Field Tool (MFT). La comunicación con los EP es por RF y con los LDC/GW por puerto serie.

*Nota: 1. El GW no se incluye en este trabajo, aunque el autor también intervino en el desarrollo de su firmware, debido a su estrecha similitud con el LDC.*

*Nota: 2. El firmware del interfaz radio RFI del HHT está incluido en el del LDC y por ello no hay un apartado propio para él en este trabajo.*

La Figura 2 contiene fotos del EP y el LDC. En la Figura 3 se puede ver un contador de gas de la marca Itron y el mismo contador con un EP instalado.

### 1.3.2 **Protocolos de comunicación**

Los protocolos de comunicación elegidos en las especificaciones del consorcio, a partir de los cuales se han de definir los nuevos mensajes y estructuras de datos como nuevos estándares, son los siguientes:

- **Parte RF**: WM-Bus y OMS.
- **Parte PLC**: PRIME y DLMS/COSEM.

También se requiere comunicación local con los equipos mediante el HHT. En el capítulo 2 se profundiza en estos protocolos.



Figura 2. EP (izda.) y LDC (dcha.).



Figura 3. Contador de gas marca Itron (izda.) y EP montado en contador de gas (dcha.).

### 1.3.3 Funcionalidades

Las funcionalidades de los equipos se pueden dividir en comunes, propias del LDC y propias del EP. A continuación se describen brevemente, según acuerdos de confidencialidad.

Las funcionalidades comunes son:

- Bajo consumo: los equipos deben ser diseñados desde una perspectiva de bajo consumo, especialmente el EP que funciona con pilas, para el que se piden varios años de autonomía.
- Comunicación RF: mediante el nuevo protocolo desarrollado, a la frecuencia 868 MHz.
- Cobertura RF: se requiere una cobertura de 200 m en exterior y 30 m en interior.
- Seguridad: implementada mediante AES 128.
- Actualización de firmware: debe permitirse la telecarga de los equipos.

Para el EP algunos de los requisitos básicos son:

- Lectura: detección correcta de la medición del consumo en el contador de gas.
- Alarmas: implementación de las siguientes alarmas:
  - Detachment: detecta si el EP ha sido desmontado del contador de gas.
  - Fraud: por si se coloca un imán que detenga la ruleta de conteo del contador de gas.
  - Low Power: cuando la alimentación (pila) está por debajo de un umbral establecido.
  - High Flow: si se da un número mínimo de pulsos por unidad de tiempo, determinado según especificaciones, se considera alarma.
- Registro de datos: se deben guardar los valores de lectura, con sus correspondientes marcas de tiempo (timestamp), de los mensajes de informe diario, así como todas las alarmas, permitiendo su posterior recuperación.
- Gestión de red: programación de las operativas de red necesarias, que se describen en los mensajes RF del apartado 2.1.
- Indicación luminosa: mediante LED se indicarán distintos estados del equipo.

El LDC debe reunir las siguientes funcionalidades:

- Comunicación local por puerto serie: por medio de un protocolo desarrollado por Atmel (ver apartado 2.3).
- Comunicación PLC: bajo protocolo PRIME y con los nuevos objetos DLMS/COSEM creados.
- Alarmas/Eventos: implementación de las siguientes alarmas:
  - EP instalados:
    - No comunicación en 7 días.
    - 5 eventos en una hora.
    - Registro de alarmas sobrescrito en menos de 35 días.
    - Periodo de NRTOD inferior a 60 min.
  - Propio LDC:
    - Reset.
    - Configuración de fecha y hora.
    - Error en la configuración de fecha y hora.
    - Error interno (acceso a memorias o al módulo de radio).
    - Actualización de firmware.
    - Error en la actualización de firmware.
- Registro de datos: almacenamiento tanto de los datos recibidos de los EP que tiene instalados (lecturas y alarmas), como de las alarmas/eventos propios.
- Gestión de red: debe ser capaz de gestionar los EP instalados en él (enlace RF) a la par que atender las solicitudes recibidas por los SSI (enlace PLC).
- Indicación luminosa: mediante LED se indicarán distintos estados del equipo.

## 1.4 Estado del Arte

La formación del consorcio por parte de la compañía energética, se deriva de la decisión de la Comisión Nacional de Energía (CNE) [2] de crear una nueva regulación para la gestión remota de contadores de gas. Atmel fue invitada a entrar en el consorcio por haber participado en la creación de un estándar abierto e interoperable en protocolos de comunicación PLC, para la telegestión de los contadores de la red eléctrica, llamado PRIME.

El estudio del estado del arte ya había sido realizado por el consorcio antes de la llegada del autor de este TFM al proyecto, definiéndose en base a éste las especificaciones del consorcio ya mencionadas. Por ejemplo, basándose en la experiencia de sus miembros, el consorcio decidió utilizar los protocolos de comunicación inalámbrica WM-Bus y OMS, mientras que se decantó por PRIME y DLMS/COSEM para las comunicaciones por PLC, en ambos casos por su buena acogida en Europa.

No obstante, el autor ha considerado interesante incluir a continuación un breve repaso de diversas iniciativas piloto relacionadas con telegestión energética, cuya información detallada no es de dominio público.

En general, conviene señalar que en los últimos tiempos se han propuesto numerosas soluciones para implementar redes de metrología o Advanced Metering Infrastructure (AMI), que soporten sistemas de Automatic Meter Management (AMM) y Automatic Meter Reading (AMR). En España, las principales compañías de suministro energético disponen de sistemas de telegestión para electricidad, como Gas Natural Fenosa (GNF) [3] e Iberdrola [4] con PRIME [13] [14] y Enel Endesa [5] con METERS AND MORE [15], ambas sobre PLC. En lo referente a sistemas AMM/AMR para gas, GNF realizó un proyecto piloto en España utilizando tecnología ZigBee (2.4 GHz) con la empresa Nuri Telecom [16], aunque existen más proyectos en el extranjero. Centrándonos en Europa, desde hace unos años el grupo ruso Gazprom [6] ofrece servicios de smart energy para electricidad y gas (GPRS); British Gas [7], en el Reino Unido, también ha desplegado redes AMI para gas; en Suecia, la empresa E.ON [8] lleva varios años proporcionando servicios de metrología de gas por GPRS; la compañía francesa GrDF [9] realizó cuatro proyectos piloto para metrología de gas en 2011, usando cabezales lectores comerciales y comunicación en las bandas 169, 433 y 868 MHz; también en Francia, la compañía ERDF [10] lanzó el proyecto Linky [17] para domótica y telemetría (incluido gas) por GPRS; otra empresa francesa, Lyonnaise des Eaux [11], llevó a cabo un proyecto piloto para AMM de gas por RF en 169 MHz y GSM/GPRS; en Malta la empresa Enemalta [12] ha promovido varios proyectos de smart energy, tanto para gas [18] como para metrología de agua y electricidad [19].

Nuevos estándares como el IEEE802.15.4g, seguido principalmente en Estados Unidos, que formalizan conceptos tales como las Smart Utility Networks (SUN), contribuyen al asentamiento de las bases para el desarrollo de redes AMI [20]. La plataforma OpenNode [21], define los nodos y enlaces que conforman una red smart grid genérica, con objeto de establecer una nomenclatura común para este tipo de proyectos. En el marco de la Comunidad Europea se sitúan el proyecto OPEN Meter [22], presentado en 2011, que recoge un amplio conjunto de estándares públicos y abiertos sobre redes AMI y el proyecto ADDRESS (2008-2013) [23] que reúne las redes de distribución de demanda activa para facilitar su explotación por las distintas compañías.

En cuanto a las tecnologías utilizadas, lógicamente, las comunicaciones PLC son las que tienen mayor acogida en el sector, por el gran ahorro que suponen, utilizando distintos protocolos, como PRIME, METERS AND MORE o G3, con modulaciones como OFDM, S-FSK o B-PSK. DLMS/COSEM es el protocolo de aplicación más común. También son importantes las comunicaciones por banda ancha por TCP/IP. De hecho, el grupo IETF [24] propone soluciones íntegramente IP, tanto para enlaces cableados como inalámbricos. No obstante, en la parte RF existen múltiples alternativas, como WM-Bus [28], OMS [29], DSMR [31], GSM/GPRS/UMTS, ZigBee [25], Bluetooth [26], WiFi [27], etc., sobre las que no se considera oportuno profundizar en este trabajo.



Para concluir, indicar que la importante innovación de este proyecto consiste en que el sistema desarrollado en él constituye la base para la inmediata estandarización de la telegestión de los contadores de gas en Europa, siguiendo la iniciativa de la CNE.

## **1.5     Planificación**

La planificación del presente trabajo se detalla en el ANEXO A, incluyendo fases y tareas acometidas, así como un cronograma de las mismas.



## 2 PROTOCOLOS DE COMUNICACIÓN

En este capítulo se comentarán los nuevos protocolos de comunicación desarrollados en el proyecto para telegestión de gas, sobre radiofrecuencia (RF) y Power Line Communications (PLC) respectivamente, hasta el punto permitido por cuestiones de confidencialidad. También se hace mención a un protocolo de Atmel utilizado para comunicación local por puerto serie.

Es importante señalar que los nuevos protocolos deben permitir la total interoperabilidad entre los distintos equipos desarrollados por cada uno de los miembros del consorcio.

### 2.1 Radiofrecuencia

La implementación de mensajes de telegestión de gas para su transmisión vía RF está basada en la norma M-Bus para telegestión [28] y en la especificación OMS [29]. La frecuencia de transmisión especificada es 868 MHz.

La norma EN 13757 elaborada por el Comité Técnico CEN/TC 294 *Sistemas de comunicación para medidores y telelectura de contadores*, lleva por nombre *Lectura remota de contadores y sus sistemas de comunicación*. Dividida en cuatro partes, define las capas física, enlace y aplicación del protocolo M-Bus para telemetría, tanto en transmisión por cable (par trenzado) como por RF.

OMS consiste en una especificación alemana (OMS Group) centrada en sistemas de lectura de contadores automática, en la que se define el concepto de MUC como concentrador de datos provenientes de distintos tipos de contadores (luz, gas, agua, etc.), para transmitirlos por diversos medios hacia los centros de gestión. Basado en M-Bus, principalmente en lo referente a dispositivos alimentados por pilas o baterías, concreta numerosos puntos relativos a las capas de enlace y aplicación.

En el proyecto se han definido los mensajes y campos de la capa de aplicación [30] necesarios para dar soporte a los requisitos del sistema AMM de gas especificado por el consorcio, en el enlace RF entre Local Data Concentrator (LDC), Endpoint (EP) y Hand Held Terminal (HHT), a partir de la norma M-Bus directamente y de las ampliaciones realizadas en OMS. En esta memoria sólo es posible enumerar las secuencias que incluyen los diferentes tipos de mensajes implementados:

- MAC Allocation: asignación de identificador WM-Bus a un EP mediante un HHT.
- Installation/Reinstalation: para instalar/reinstalar un EP en un LDC. Incluye mensajes para comunicación con HHT y LDC. Soporta self-discovering en la red.
- Return to Factory Mode: para desinstalar un EP. Comunicación con HHT.
- Daily Report: envío de la lectura diaria a determinada hora (configurable) con la marca temporal (timestamp). Entre EP y LDC.
- NRTOD: para posibilitar tablas de perfiles horarios y permitir el envío de comandos desde el LDC, se establece la transmisión de mensajes periódicos de lectura con timestamp, normalmente a las horas en punto (periodo configurable). Entre EP y LDC.
- Time&Period Setup: el LDC envía modificación de la hora del Daily Report y el periodo del NRTOD al EP.
- Alarm: envío del estado de alarma con su correspondiente timestamp del EP al LDC.

- Logs Retrieval EP: recuperación de los registros de lectura y alarmas del EP bien con un HHT bien desde el LDC.
- FU: conjunto de secuencias para procesos de actualización de firmware. Entre EP y LDC.

En general, el autor ha colaborado en la definición de todas las secuencias, aunque las de MAC Allocation, Daily Report, NRTOD, Time&Period Setup y Alarm ya se encontraban bastante concretadas por el consorcio antes de su llegada al proyecto. Sin embargo, las aportaciones realizadas por el autor en el resto de secuencias de mensajes fueron determinantes, hasta el punto de que finalmente se incluyeron sus propuestas de forma casi exacta.

## 2.2 Power Line Communications

La compañía energética ya utilizaba DLMS/COSEM [32] [33] y PRIME [13] sobre PLC en sus sistemas de telegestión de electricidad, por lo que se especificó la definición de nuevos objetos COSEM para gas a utilizar en las comunicaciones PLC entre el LDC y el Gestor del Centro de Transformación (GCT).

Las comunicaciones PLC con el protocolo PRIME fueron desarrolladas desde sus inicios por Atmel Spain (anteriormente ADDSemi), y no forman parte del objeto del presente proyecto.

DLMS/COSEM es una especificación para modelos de datos y protocolos en operaciones de telemetría. DLMS consiste en la generalización de un modelado abstracto de entidades de comunicación, mientras que COSEM establece las reglas, según estándares existentes, para el intercambio de datos con contadores de energía.

Al igual que en el caso de RF, sólo se ha permitido mostrar en esta memoria un listado de los grupos de nuevos objetos implementados [34] (siempre comunicación LDC-GCT):

- Management: conjunto de objetos para gestión de red a nivel DLMS.
- Reading: secuencias destinadas al envío de datos de lectura de los EP.
- Alarm: incluye el envío de registros de alarmas tanto de los EP como del propio LDC.
- Setup: para configuración y gestión de red a nivel WM-Bus.

En este caso, la contribución del autor ha sido menor, ya que casi todo el trabajo estaba hecho cuando se incorporó al proyecto, limitándose a la depuración de errores, así como al aporte de rigor y coherencia, con pequeñas modificaciones y ampliaciones.

## 2.3 Puerto serie

Hay que mencionar que para la comunicación local por puerto serie (USB) entre la MFT y el HHT o el LDC, otros miembros del equipo Atmel desarrollaron un protocolo propio llamado Backdoor [40].

### 3 **FIRMWARE**

En este capítulo se describen detalladamente las estructuras de la programación empotrada llevada a cabo tanto en el microcontrolador (MCU) AT91SAM3SD8C [35] del Local Data Concentrator (LDC) como en el MCU ATmega328P [36] del Endpoint (EP). Ambos firmwares son versión v7.

Para una mejor comprensión del firmware se recomienda consultar previamente el ANEXO B, donde se presentan los esquemas hardware de ambas placas (LDC y EP).

En el ANEXO C se incluyen las herramientas de desarrollo firmware utilizadas, tanto para el LDC como para el EP.

#### 3.1 **Local Data Concentrator**

En el LDC, la arquitectura del firmware comienza por un workspace dividido en dos proyectos independientes: Bootloader y Application. El primero es necesario para los procesos de actualización de firmware (FU) y el segundo incluye el código para implementar el resto de funcionalidades.

El Bootloader es un programa sencillo que no se ejecuta en tiempo real, sino sólo al arrancar el dispositivo o cuando se produce un reset, encargándose de lanzar la ejecución del código de Application. Se ha programado en lenguaje C++.

La Application sí requiere ejecución en tiempo real. Es importante señalar que inicialmente se decidió una primera estructuración del firmware basada en la técnica Foreground/Background [37], pero posteriormente, viendo la complejidad que el firmware estaba alcanzando, esta primera estructura se introdujo como una tarea dentro del sistema operativo en tiempo real (RTOS) FreeRTOS [37][38]. En adelante se hará referencia a las funcionalidades incluidas en dicha tarea como BaseApp.

El firmware de Application está dividido en los módulos siguientes:

- **Módulo Startup**: el primero en ejecutarse tras el arranque con el Bootloader. Incluye la inicialización del hardware y de FreeRTOS con las tareas programadas.
- **Módulo COSEM**: contiene los objetos COSEM.
- **Módulo PLC**: es una implementación de la pila de comunicación PRIME, usando el interfaz AddUSI.
- **Módulo Image Transfer**: recoge los objetos COSEM para la actualización de firmware y el código necesario para la activación de una nueva imagen.
- **Módulo BaseApp**: gestiona la comunicación RF mediante el protocolo WM-Bus, el acceso por backdoor (USB), así como los canales disponibles para la instalación de los EP y sus correspondientes datos, incluida la telecarga.

El código de la Application presenta partes en C y otras en C++. Los módulos Startup, COSEM, PLC e Image Transfer están programados en C++, aunque utilizan librerías externas programadas en C, mientras que el BaseApp se programó completamente en C.

Además, el proyecto Application incluye las siguientes librerías externas:

- libchip y libboard: aplicaciones de microcontrolador de bajo nivel.
- FreeRTOS: código del sistema operativo.
- AddUSI: interfaz de puerto serie para comunicar con el ADD1021 como modem PRIME.
- COSEM Server: librería que contiene la capa de aplicación COSEM, los objetos COSEM y su funcionalidad. Programada en su totalidad por una subcontrata externa.
- FTL: librería de la Flash Translation Layer (FTL) que permite el nivelado de la flash externa.

El autor del presente trabajo se ha encargado principalmente de la estructuración de la arquitectura del firmware, la interacción con el hardware y la supervisión de la programación, a la que contribuyó con importantes aportaciones puntuales y de forma profunda tanto en la comunicación con el módulo de radio como en la implementación del protocolo WM-Bus. En cuanto a las librerías externas, no ha programado ninguna, limitándose a su configuración y adaptación para su integración en el proyecto, en colaboración con el resto del equipo firmware.

### 3.1.1 Esquema general

La Figura 4 muestra el esquema general del firmware, proporcionando una visión global de los distintos módulos y tareas que conforman el LDC (formas rectangulares). El esquema también incluye los periféricos del microcontrolador (formas ovaladas) y de la placa (formas romboidales) utilizados.

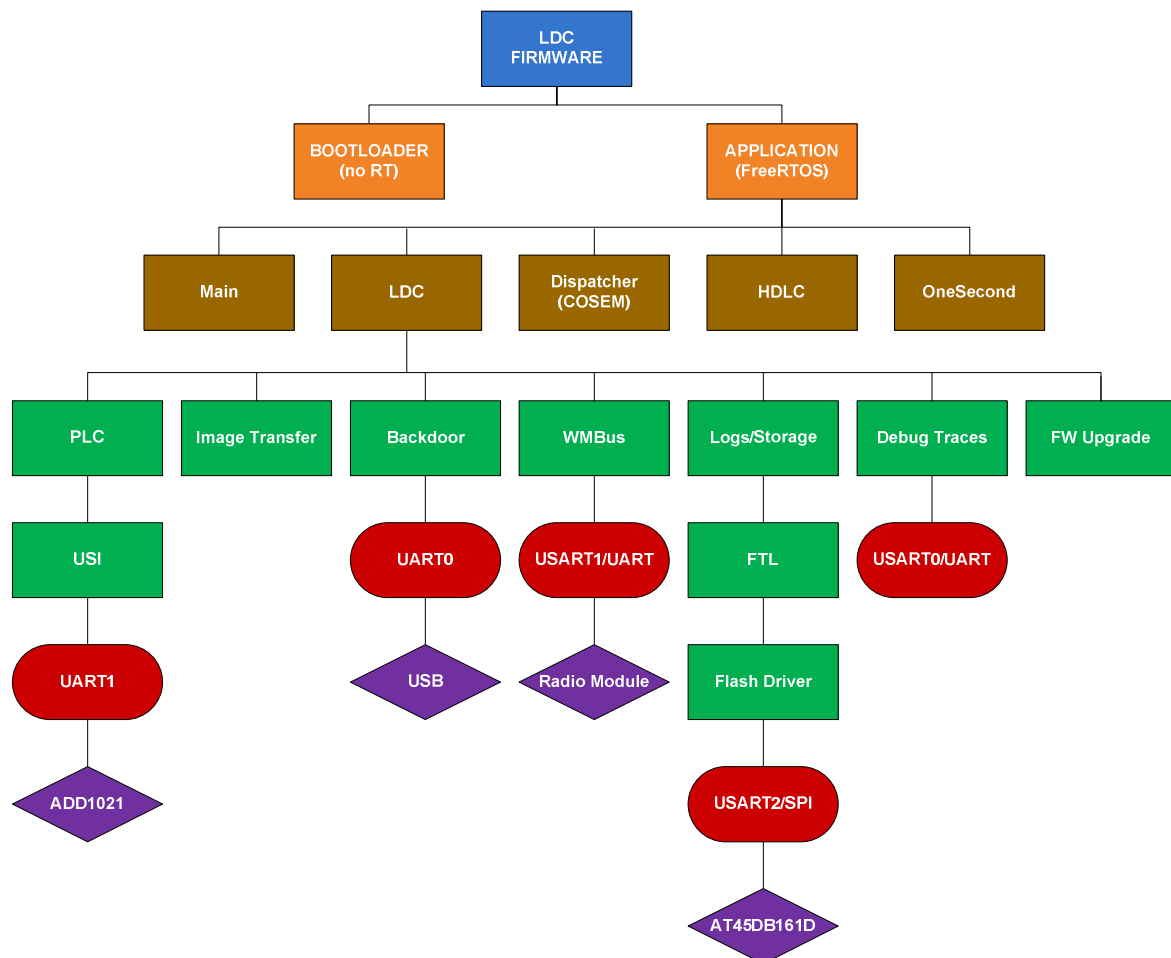


Figura 4. Esquema FW del LDC.

### 3.1.2 **Bootloader**

Como ya se ha dicho el Bootloader es un programa sencillo que no se ejecuta en tiempo real, sino que sólo se inicia en cada arranque de la placa o cuando se produce un reset. Por lo tanto, se encarga de lanzar la Application.

En primer lugar activa el watchdog con un timeout de 3 segundos. Si la Application no se ejecuta correctamente y por lo tanto el watchdog no es alimentado, hará un reset y el Bootloader se ejecutará de nuevo. El microcontrolador cuenta con una memoria flash de doble banco para posibilitar las actualizaciones de firmware. Al comienzo de ambos bancos se carga el Bootloader. Esta parte no se actualiza, sólo el área de Application se sobrescribe cuando se lleva a cabo una actualización, durante la que el firmware realiza un reset y se ejecuta el Bootloader, que se encarga de cambiar el banco activo y trata de poner en marcha la nueva imagen de Application. Si el Bootloader detecta que la nueva imagen no se ha ejecutado correctamente, conmutará de vuelta a la antigua imagen.

### 3.1.3 **Application**

#### 3.1.3.1 **Sistema en tiempo real**

El código de Application implementa un sistema operativo en tiempo real (RTOS) para dispositivos embebidos, llamado FreeRTOS [38]. Se trata de un RTOS de software libre, muy simple y de rápida ejecución, que se adecúa perfectamente a las necesidades del proyecto. El código se incluye como librería externa dentro del workspace. Las tareas o hilos implementados en FreeRTOS son las siguientes:

- Tarea Main: es la tarea principal, desde donde se realizan todas las inicializaciones y se lanzan el resto de tareas. Tras la secuencia de inicio esta tarea se detiene. Para ahorrar recursos, su contexto es traspasado a la tarea Dispatcher.
- Tarea LDC: esta tarea ejecuta el bucle de la estructura original (Foreground/Background), es decir, el módulo BaseApp, que incluye las funcionalidades básicas, así como el código para la gestión de la comunicación PLC mediante el interfaz AddUSI y el de Image Transfer.
- Tarea Dispatcher: todas las solicitudes COSEM recibidas por cualquiera de las capas de aplicación COSEM son tratadas en esta tarea.
- Tarea HDLC: esta tarea se encarga del protocolo HDLC para comunicación local (USB) con protocolo DLMS/COSEM. Se activa sólo si la comunicación por backdoor está deshabilitada, ya que ambos protocolos utilizan el mismo recurso hardware (UART0 con interrupción en recepción) para la comunicación serie.
- Tarea OneSecond: aquí se ejecutan cada segundo las acciones relacionadas con COSEM programadas para ese segundo. Sus resultados son utilizados por la tarea Dispatcher. Sujeta a la interrupción del TC1.

#### 3.1.3.2 **Módulo Startup**

Este módulo contiene el código de inicialización del hardware y el firmware, necesario para empezar a ejecutar la aplicación.

El primer paso en esta inicialización es realizado por las librerías `libchip_sam3sd8` y `libboard_sam3s-ek`, después de que se llame a la función `main()` desde el módulo `startup`. La secuencia de inicialización es la siguiente:

1. Comprobar que la alimentación de 230 V es correcta. De lo contrario se entra en el modo dormir (backup mode). Para ello se utiliza el ADC.
2. Realizar la inicialización del hardware: NVIC (Nested Vectored Interrupt Controller), temporizadores o timers, trazas para depuración, etc. También se lleva a cabo la inicialización del puerto serie HDLC si las opciones de precompilación se han configurado para ello (backdoor deshabilitada).
3. Ejecución de la inicialización de la estructura original.
4. Comenzar la tarea Main y el planificador de FreeRTOS (scheduler).
  1. Carga de datos persistentes.
  2. Descubrir el nombre del dispositivo lógico e inicializar la pila PRIME.
  3. Comenzar el gestor de eventos de BaseApp en su correspondiente hilo.
  4. Crear los dispositivos lógicos COSEM y registrarlos en el planificador.
  5. Inicializar el módulo Image Transfer.
  6. Inicializar la tarea HDLC, si éste está activado (no backdoor).
  7. Comenzar la tarea de gestión del reloj.
  8. Como último paso, lanzar la tarea Dispatcher sobre el contexto de la tarea Main.

### 3.1.3.3 Módulo COSEM

Programado en su totalidad por una subcontrata externa, este módulo contiene todos los objetos COSEM, excepto los relacionados con el proceso de telecarga, incluidos en el módulo Image Transfer. A continuación se presentan los conjuntos de objetos COSEM reunidos aquí:

- Objetos Management: normalmente ubicados en el dispositivo lógico 1.
- Objetos WM-Bus: pueden situarse en cualquier dispositivo lógico, en nuestro caso el 1.

La implementación de los objetos COSEM está basada en la librería COSEM Server. Esta librería proporciona diversas funcionalidades para facilitar la creación de objetos COSEM. En algunos casos el objeto COSEM completo ya está programado en la librería y puede ser usado sin realizar cambios. Para otros objetos existe una implementación que debe ser configurada y ampliada según el proyecto. También hay objetos cuyo esqueleto de clase de interfaz es un buen punto de partida, o si el objeto pertenece a una clase de interfaz personalizado, puede ser desarrollado usando la clase base de todos los objetos COSEM. Si varios objetos presentan estructuras similares, es posible crear grupos. Estos grupos son capaces de servir solicitudes COSEM dirigidas a cualquiera de los objetos del grupo sin necesidad de crear una instancia para cada objeto.

En el conjunto de objetos creados en este apartado se engloban los que conforman el nuevo estándar DLMS/COSEM para gas desarrollado en el marco de este proyecto.

La tarea FreeRTOS que gestiona este módulo es la Dispatcher.



### 3.1.3.4 Módulo PLC

El módulo PLC se encarga de la monitorización del microcontrolador ADD1021 en el que está implementado el protocolo PRIME para la comunicación PLC, así como de mantener la conexión LLC 4-32 sobre la que corre la capa de aplicación COSEM, gestionada desde el AT91SAM3SD8C.

Para ello, el módulo utiliza el interfaz AddUSI, que permite la comunicación de bajo nivel entre el AT91SAM3SD8C y el ADD1021, usando la UART1 con interrupción por recepción. La parte servidor del interfaz está ubicada en el AT91SAM3SD8C, mientras que el ADD1021 alberga la parte cliente. El interfaz hace uso del protocolo USI de Atmel Spain [39].

Para la conexión 4-32, el micro intenta detectar y recuperar la conexión bien tras un arranque bien si ésta se ha perdido. La detección implica llevar a cabo un polling (encuesta) del estado de la conexión LLC y el estado de registro de la capa MAC, a través del AddUSI. Si la verificación falla, la pila reinicia el proceso de conexión desde el estado correspondiente. Los pasos son los siguientes:

1. El modem realiza el registro de capa MAC automáticamente. La pila espera consultando mediante encuesta hasta que la capa MAC indica que se ha registrado y se encuentra operativo.
2. Si se produce desconexión, la pila envía una solicitud de establecimiento de 4-32 y se queda a la espera de una respuesta. Este proceso se repite indefinidamente si expira el tiempo límite establecido.
3. Si está conectado, el estado de dicha conexión es verificado periódicamente. Ante una indicación de desconexión, se vuelve al paso primero.

Para ahorrar recursos la pila PRIME es ejecutada periódicamente en la tarea LDC, en vez de en una tarea propia. La conexión con la tarea Dispatcher y las capas COSEM se lleva a cabo mediante una sencilla clase que reenvía las solicitudes de datos desde COSEM a la pila PRIME.

### 3.1.3.5 Módulo Image Transfer

En este módulo se recogen los objetos COSEM necesarios para llevar a cabo procesos de telecarga o actualización de firmware, así como el código necesario para la activación de una nueva imagen. Al igual que el módulo COSEM, el módulo Image Transfer ha sido programado por una subcontrata externa.

Estos objetos, ubicados en la librería COSEM Server (ver apartado 3.1.3.3), implementan las clases encargadas de la carga de la nueva imagen en el banco libre del micro (donde no corre el código actual), realizar las comprobaciones de datos oportunas, acceder al Bootloader para llevar a cabo la conmutación entre bancos y pasar de la imagen vieja a la nueva, controlar el arranque de la nueva imagen y verificar que su versión es la correcta. Obviamente, el Bootloader tiene que estar presente en ambos bancos del micro para poder realizar actualizaciones de firmware independientemente del banco que se esté ejecutando.

La gestión del proceso de telecarga anteriormente descrito, tanto desde el Bootloader como desde la Application ha sido programada siguiendo el estándar DLMS, mediante una compleja máquina de estados.

El módulo Image Transfer se gestiona por medio de la tarea LDC de FreeRTOS.

### 3.1.3.6 Módulo BaseApp

Como ya se ha comentado anteriormente, el módulo BaseApp es el encargado de controlar diversos submódulos, como son la comunicación RF mediante el protocolo WM-Bus, el acceso externo por backdoor (USB), así como los canales disponibles para la instalación de Endpoints (EP) y sus correspondientes datos, incluida la telecarga (FU). También se incluyen en este módulo el conjunto de trazas definidas para pruebas. A continuación se desglosan los submódulos implementados y se explica la gestión de las tareas asociadas.

#### 3.1.3.6.1 Submódulo WM-Bus

Este submódulo, programado íntegramente por el autor de este TFM, se ocupa de implementar la comunicación radio bajo el protocolo WM-Bus con los EP. El código contiene las funciones de configuración del módulo de radio y código de bajo nivel para controlar dicho módulo. Asimismo, en esta parte del proyecto se definen los mensajes creados para el nuevo estándar WM-Bus de telegestión de gas objeto del presente proyecto [30].

Dado que el LDC siempre tiene que estar a la escucha de nuevos mensajes radio, su módulo radio nunca duerme ni se apaga, estando sujeta la recepción de estos mensajes a la rutina de interrupción de la USART1/UART, que constituye el interfaz de comunicación entre el micro AT91SAM3SD8C y el módulo de radio.

Como la capa de aplicación de los mensajes WM-Bus va encriptada, el módulo soporta encriptación AES 128.

#### 3.1.3.6.2 Submódulo Backdoor

Se trata del submódulo encargado de la configuración y gestión del LDC vía USB desde una aplicación PC, la MFT. Para ello, implementa el protocolo del mismo nombre definido por Atmel Spain para este proyecto [40].

El submódulo se activa mediante una etiqueta de precompilación, desactivando con ello la comunicación por HDLC. Estos dos protocolos son incompatibles ya que ambos emplean la misma UART0, cuya rutina de interrupción indica la recepción de mensajes.

#### 3.1.3.6.3 Submódulo Logs/Storage

Este submódulo se ocupa de gestionar todos los accesos a la memoria flash externa AT45DB161D. Sus funciones son llamadas desde cualquier módulo que requiera almacenamiento en memoria fija, por ejemplo para el registro de datos de configuración, datos de lectura o alarmas enviados por los EP instalados o alarmas propias del LDC.

Su principal cometido es evitar un excesivo desgaste de la flash externa. Para conseguirlo, determinados datos son almacenados en buffers RAM antes de ser enviados a la librería FTL de nivelado, de manera que se configuren como páginas de la flash lo más completas posible, limitando así el número de escrituras necesarias.

#### 3.1.3.6.4 Submódulo Driver de la flash externa

Aparte de la configuración de la librería FTL para el nivelado de la flash externa AT45DB161D, fue preciso desarrollar el driver de bajo nivel de acceso a la misma. El driver utiliza la USART2/SPI.

#### 3.1.3.6.5 Submódulo FW Upgrade

Se trata de la parte del código que define la máquina de estados para la telecarga de los EP instalados en el LDC, que debe realizarse vía radio. Está basada en la estructura utilizada en PRIME, aunque fueron necesarias modificaciones importantes por las diferentes peculiaridades del medio físico y el protocolo de comunicación, que en este caso son RF bajo WM-Bus en vez de PLC bajo PRIME.

*Nota: No confundir esta actualización de firmware para WM-Bus con la implementada en el módulo Image Transfer para DLMS/COSEM.*

#### 3.1.3.6.6 Submódulo Debug Traces

Con objeto de posibilitar distintos niveles de comprobación y depuración de código, se creó un completo sistema de trazas, con salida por la USART0/UART, que pueden ser recibidas en un PC por puerto serie para su visualización en un terminal.

#### 3.1.3.6.7 Gestión de tareas

La tarea de FreeRTOS que gestiona todos estos submódulos, aparte de la comunicación PLC mediante el interfaz AddUSI y actualización del firmware del equipo, es la tarea LDC. Incluye la estructura original BaseApp basada en la técnica de implementación de tiempo real Foreground/Background. Es decir, dentro de esta tarea se ha programado un bucle con dos planos de subtareas, dividiéndolas en menos prioritarias (Plano 1º) y en más prioritarias (Plano 2º), estas últimas sujetas a rutinas de interrupción.

A continuación se detallan todas estas tareas.

##### Tareas Plano 1º

- Tarea taskAliveRun: comprueba la alimentación del MCU y ejecuta el modo dormir si es necesario. Si es así, configura los pines IO para minimizar el consumo y conmuta la alimentación a la pila externa, manteniendo en todo caso la fecha y hora (tarea de Plano 2º taskRTC) y alimentando el watchdog.
- Tarea taskLiveStateRun: se encarga de detener de forma adecuada todas las tareas si se ha lanzado el modo dormir.
- Tarea taskMBUSRun: gestiona las comunicaciones por RF, dependiendo para la recepción de la tarea de Plano 2º taskMBusRx. Además, realiza el procesamiento de los mensajes radio mediante el submódulo WMBus y la gestión de red, que incluye la instalación de nuevos EP y el mantenimiento de los canales existentes.
- Tarea taskPLCRUn: controla la comunicación PLC mediante el interfaz AddUSI (módulo PLC). Vinculada a las tareas de Plano 2º taskUSIRx y taskUSI.

- Tarea taskBackDoorRun: monitoriza las comunicaciones por Backdoor por medio de submódulo del mismo nombre y haciendo uso de la información proporcionada por la tarea de Plano 2º taskBackDoorRx.
- Tarea taskImageActivation: si en un proceso de actualización de firmware del LDC se ha cargado correctamente la nueva imagen, esta tarea se encarga de resetear el equipo para que el Bootloader lleve a cabo la conmutación de banco y se pase a ejecutar la nueva imagen de la Application. Por lo tanto, gestiona el módulo Image Transfer.
- Tarea fuMasterProcess: gestiona la máquina de estados del submódulo FW Upgrade, para la telecarga de los EP.
- Tarea taskLogsRun: comprueba si se ha desinstalado algún EP y limpia los logs del canal correspondiente.

### Tareas Plano 2º

- Tarea taskRTC: actualiza la fecha y hora cada segundo por medio de la interrupción RTC. Estos parámetros son utilizados por diversas tareas, tanto de FreeRTOS como del Plano 1º tratado aquí.
- Tarea taskTicks: la interrupción del TC0 gestiona dos contadores de ticks cada milisegundo, incrementándolos según distintas condiciones, los cuales son usados por distintas tareas de Plano 1º.
- Tarea taskMBusRx: sujeta a la interrupción de la USART1/UART, guarda cada byte recibido en un buffer para su posterior tratamiento por la tarea de Plano 1º taskMBUSRun.
- Tarea taskUSIRx: vinculada con la interrupción de la UART1, guarda cada byte recibido en un buffer para su posterior tratamiento por la tarea de Plano 1º taskPLCRun.
- Tarea taskUSI: reconfigura cada milisegundo determinados parámetros del USI y emplea flags para indicar estados a la tarea de Plano 1º taskPLCRun. Depende de la interrupción del TC0.
- Tarea taskBackDoorRx: sujeta a la interrupción de la UART0, guarda cada byte recibido en un buffer para su posterior tratamiento por la tarea de Plano 1º taskBackDoorRun.

## **3.2 Endpoint**

El firmware del EP, aunque más simple que el del LDC, también incluye en su workspace un proyecto para el Bootloader y otro independiente para la Application. Como ya se explicó en el apartado 3.1, el primero es necesario para los procesos de actualización de firmware (FU) y el segundo incluye el código para implementar el resto de funcionalidades.

Cabe recordar que el Bootloader es un programa sencillo que no se ejecuta en tiempo real, sino sólo al arrancar el dispositivo o cuando se produce un reset, encargándose de lanzar la ejecución del código de Application.

Por otro lado, la Application sí requiere ejecución en tiempo real. Para conseguir este requisito, la arquitectura del firmware sigue la técnica Foreground/Background [37],

El firmware de Application está dividido en los módulos siguientes:

- Módulo Startup: el primero en ejecutarse tras el arranque con el Bootloader. Incluye la inicialización del hardware y nos introduce en el bucle de scan, donde serán tratadas las distintas tareas.

- Módulo WMBus: contiene las funciones de configuración del módulo de radio y de gestión de la comunicación mediante el protocolo WM-Bus.
- Módulo Logs/Storage: tiene como cometido el almacenamiento de logs y datos persistentes en la memoria EEPROM interna.
- Módulo FW Upgrade: código necesario para la actualización de firmware y la activación de una nueva imagen.
- Módulo extEEPROM: consiste en el driver de la memoria EEPROM externa, que se usa en procesos de telecarga.
- Módulo BaseApp: se encarga de monitorizar las señales enviadas por los sensores de la placa y su procesamiento, así como de la gestión de red.

Tanto el código del Bootloader como el de la Application se han programado íntegramente en lenguaje C, labor llevada a cabo en su totalidad por el autor de este trabajo, junto con la previa estructuración de la arquitectura del firmware.

### 3.2.1 Esquema general

La Figura 5 muestra el esquema general del firmware, proporcionando una visión global de los distintos módulos y tareas que conforman el EP (formas rectangulares). El esquema también incluye los periféricos del microcontrolador (formas ovaladas) y de la placa (formas romboidales) utilizados.

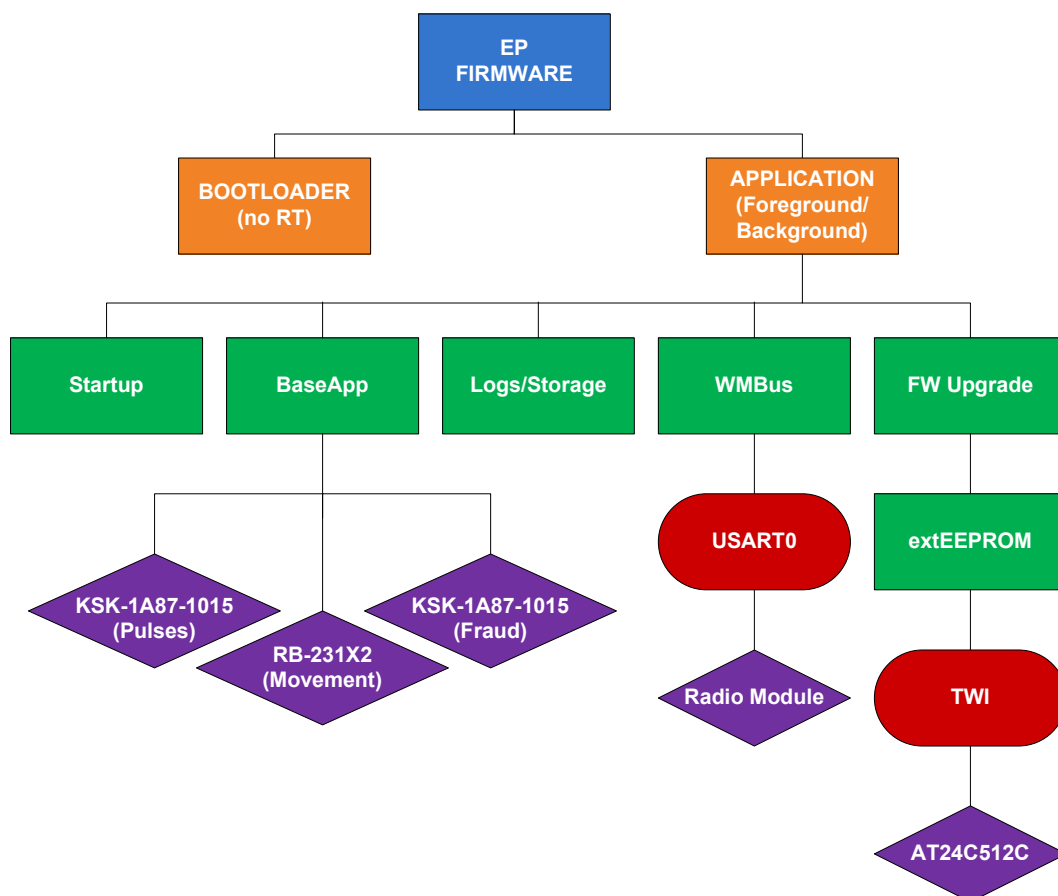


Figura 5. Esquema FW del EP.

### 3.2.2 Bootloader

Como ya se ha dicho el Bootloader es un programa sencillo que no se ejecuta en tiempo real, sino que sólo se inicia en cada arranque de la placa o cuando se produce un reset. Por lo tanto, se encarga de lanzar la Application.

Aparte de habilitar el watchdog, su función principal es detectar si se trata de un primer arranque, en cuyo caso realizará una copia de la imagen de Application en el primer banco de la memoria EEPROM externa (backup), o si es un reset. En el segundo caso, el Bootloader debe comprobar si el MCU se encuentra en un proceso de telecarga y, de ser así, copiar del banco correspondiente de la EEPROM externa la imagen nueva en la flash del MCU y lanzar su ejecución. Su máquina de estados contempla los casos de ejecución errónea de la Application lanzada, volviendo a la anterior imagen (guardada en la EEPROM externa) si procede.

### 3.2.3 Application

#### 3.2.3.1 Sistema en tiempo real

El código de Application implementa ejecución en tiempo real. mediante una estructura basada en la técnica Foreground/Background [37], consistente en un bucle de scan, al que se entra tras realizar las inicializaciones necesarias, donde se tratan periódicamente diversas tareas de prioridad secundaria (plano 1º), dejando para las rutinas de interrupción las tareas de primera prioridad (plano 2º).

A continuación se desglosan ambos planos con sus correspondientes tareas.

##### Plano 1º

- Tarea Main: ubicada fuera del bucle de scan, es la primera tarea en ejecutarse, lo cual sucede sólo una vez tras ser lanzada la Application desde el Bootloader. Gestiona el módulo Startup de inicialización.
- Tarea Sleep: para cumplir con el requisito de bajo consumo, la primera tarea en ejecutarse tras entrar en el bucle de scan es la de dormir el MCU (sleep). Antes de ir a dormir se deshabilita el watchdog, se apagan los periféricos internos (ADC), se apagan los periféricos externos (radio, EEPROM y los LED) y se configura el modo de más bajo consumo que permite las interrupciones de TC2 y de IO. Estas interrupciones, sujetas a las tareas de Plano 2º RTCseg y Pulses, despiertan al MCU para que siga ejecutando el resto de tareas del bucle de scan.
- Tarea RTC: cada 8 segundos como mínimo actualiza la fecha y la hora, leyendo los segundos actualizados por la tarea de Plano 2º RTCseg.
- Tarea FUstate: consulta la máquina de estados de telecarga y realiza las operaciones necesarias, haciendo uso del módulo FW Upgrade.
- Tarea CheckAlarms: realiza la comprobación de las alarmas dentro del módulo BaseApp. Hace uso de los datos actualizados por las tareas de Plano 2º RTCseg y Pulses.
- Tarea Setup: se encarga del control de las operaciones de configuración y gestión externa mediante un Hand Held Terminal (HHT) recogidas en el módulo BaseApp.
- Tarea AlarmTx: si hay alguna alarma, inicia el registro de su log en la EEPROM interna y procede a su transmisión radio. Emplea el módulo BaseApp.

- Tarea ReportTx: gestiona la construcción y envío por radio de los mensajes con los valores de lectura, realizando el registro de logs en la EEPROM interna según proceda, dependiendo del tipo de report. Utiliza los pulsos actualizados por la tarea de Plano 2º Pulses. Hace uso del módulo BaseApp.

*Nota: La tarea Sleep no necesitaría apagar ningún periférico porque son apagados siempre después de cada uso por otras tareas, pero se apagan de nuevo como medida de seguridad de cara al consumo.*

#### Plano 2º

- Tarea RTCseg: haciendo uso de la interrupción del TC2, programada cada 8 segundos, actualiza los segundos para que la tarea de Plano 1º RTC pueda actualizar la fecha y la hora. También incrementa un contador, necesario para la detección de la alarma de High Flow (tarea de Plano 1º CheckAlarms). Si el MCU está dormido esta tarea termina la tarea de Plano 1º Sleep.
- Tarea Pulses: recoge los pulsos emitidos por el sensor reed switch KSK-1A87-1015 de lectura, a través de la interrupción de IO del pin correspondiente, y actualiza variables para la detección de la alarma de High Flow. Esta información es utilizada por las tareas de Plano 1º CheckAlarms y ReportTx. Además, si el MCU está dormido esta tarea termina la tarea de Plano 1º Sleep.

### **3.2.3.2 Módulo Startup**

Se trata del primer módulo en ejecutarse tras el arranque con el Bootloader. Se rige por la tarea Main y, antes de finalizar, nos introduce en el bucle de scan, donde serán tratadas el resto tareas.

Los pasos seguidos en este proceso son:

1. Inicialización del hardware:
  1. Configuración de los pines de cada puerto.
  2. Configuración del TC2 mediante oscilador externo para realizar la función de RTC.
  3. Configuración del ADC.
  4. Configuración de la EEPROM externa.
  5. Configuración de interrupciones.
  6. Configuración del módulo de radio.
2. Comprobación del correcto funcionamiento del TC2.
3. Comprobación de asignación de identificador (ID) WM-Bus al dispositivo. Si el EP no tiene ID, se mantendrá en un bucle infinito solicitándola a través de la radio hasta que se le asigne una. Este proceso debe realizarse al final de la cadena de producción (EOL).
4. Inicialización de parámetros de red, cargándolos desde la EEPROM interna según proceda.
5. Inicialización de parámetros de FU, cargándolos desde la EEPROM interna según proceda.
6. Comienzo del bucle de scan.

### **3.2.3.3 Módulo WMbus**

Este módulo se ocupa de implementar la comunicación radio bajo el protocolo WM-Bus con los LDC o HHT. El código contiene las funciones de configuración del módulo de radio y código de bajo nivel para controlar dicho módulo. Asimismo, en esta parte del proyecto se definen los

mensajes creados para el nuevo estándar WM-Bus de telegestión de gas objeto del presente proyecto [30].

Los módulos FW Upgrade y BaseApp utilizan este módulo para realizar la transmisión y recepción de mensajes de RF por WM-Bus. Ellos se ocupan de encender y apagar la radio cada vez que hacen uso de ella, para minimizar el consumo, a través de una etapa transistorizada conectada a la alimentación del módulo de radio.

La comunicación del MCU con el módulo de radio se lleva a cabo por la USART0 sin usar interrupciones, sino atendiendo a la recepción de mensajes por encuesta.

Como la capa de aplicación de los mensajes WM-Bus va encriptada, el módulo soporta encriptación AES 128.

#### **3.2.3.4 Módulo Logs/Storage**

El módulo Logs/Storage se encarga del registro de logs de lecturas y alarmas, así como de guardar los datos persistentes, empleando la memoria EEPROM interna. Es utilizado por el resto de módulos cuando requieren datos almacenados en memoria permanente

#### **3.2.3.5 Módulo FW Upgrade**

Se trata de la parte del código que define la máquina de estados para la telecarga del EP, proceso iniciado por el LDC en el que está instalado por RF con el protocolo WMBu. Está basada en la estructura utilizada en PRIME, aunque fueron necesarias modificaciones importantes por las diferentes peculiaridades del medio físico y el protocolo de comunicación, que en este caso son RF bajo WM-Bus en vez de PLC bajo PRIME.

La tarea que lo gestiona es la FUstate.

#### **3.2.3.6 Módulo extEEPROM**

Se trata principalmente del driver de la EEPROM externa, que sólo es usada en el primer arranque del equipo (Bootloader) o en los procesos de actualización del firmware (Bootloader y Application).

Aparte de las funciones de configuración, lectura y escritura, el driver utiliza la interrupción del puerto TWI (compatible con el protocolo Philips' I<sup>2</sup>C) por el que está conectada la EEPROM externa, para implementar dentro de su ISR una máquina de estados para la gestión del tráfico de datos a bajo nivel.

Este módulo es llamado desde el módulo FW Upgrade, vinculado a la tarea FUstate, por lo que no tiene tarea propia. Tampoco se ha contemplado una tarea de Plano 2º asociada a la interrupción del TWI, ya que ésta es habilitada y deshabilitada por el propio driver de la EEPROM externa con cada uso de esta memoria.



### 3.2.3.7 Módulo BaseApp

Dentro de este módulo se monitorizan los sensores de la placa, tanto de lectura de pulsos de contaje como de alarmas, así como otras señales provenientes de la placa. También se incluye el procesado de todos estos datos externos recibidos. Por último, aquí se encuentra el código que implementa la tarea de configuración y gestión de red, atendiendo solicitudes de conexión de un HHT externo. Según esto, el módulo se divide en tres submódulos, explicados a continuación.

#### 3.2.3.7.1 Submódulo Setup

Para llevar a cabo operaciones de instalación, reinstalación o desinstalación (vuelta a modo factory), así como de recuperación de registros de logs, se requiere un equipo de campo para configuración y gestión, el HHT. La comunicación con el HHT se realiza por radiofrecuencia, utilizando el protocolo WM-Bus.

La tarea Setup comprueba si un HHT quiere conectarse al EP para llevar a cabo alguno de estos procedimientos y, de ser así, se introduce en el código de procesado correspondiente.

#### 3.2.3.7.2 Submódulo Alarms

Tiene la misión de comprobar si el EP está en situación de alarma para modificar su estado y de transmitir un mensaje asíncrono, previo registro de la alarma, en el caso de que este estado cambie.

Desde la tarea CheckAlarms, se consulta por encuesta o polling el hardware ligado a cada alarma para analizar si sus valores superan los umbrales establecidos. Ya se comentó que existen cuatro tipos de alarmas, cuya detección se detalla a continuación:

- Detachment: detecta si el EP ha sido desmontado del contador de gas. Para ello se consulta el pin del MCU conectado al sensor de movimiento RB-231X2 de la placa.
- Fraud: por si se coloca un imán que detenga la ruleta de contaje del contador de gas. Lee el valor del pin del MCU por el que se recoge la señal del sensor reed switch KSK-1A87-1015 correspondiente.
- Low Power: haciendo uso del ADC, compara el valor de un pin conectado a la alimentación (pila) con el umbral establecido.
- High Flow: depende de los contadores y variables que se actualizan con las interrupciones del TC2 y la IO del sensor reed switch KSK-1A87-1015 de lectura de pulsos. Si se da un número mínimo de pulsos por unidad de tiempo, determinado según especificaciones, se considera alarma.

El registro del log de la alarma en la memoria EEPROM interna y la transmisión del mensaje asíncrono por radio sobre WM-Bus, se gestionan mediante la tarea AlarmTx.

#### 3.2.3.7.3 Submódulo Report

Para entender el funcionamiento de este módulo conviene explicar los dos tipos de mensajes de lecturas definidos:

- Mensaje Daily Report: contiene el valor de lectura hasta la hora del día fijada por la compañía energética y será utilizado por ésta para los cálculos de facturación. Tanto la lectura como la fecha y hora (timestamp) han de ser registrados (EEPROM interna).
- Mensaje NRTOD: para posibilitar tablas de perfiles horarios y permitir el envío de comandos desde el LDC, se establece la transmisión de mensajes periódicos, normalmente a las horas en punto (periodo configurable).

El submódulo Report se encarga de la construcción y envío por radio sobre WM-Bus de ambos tipos de mensaje, realizando el registro de logs en la EEPROM interna sólo si son de Daily Report. Implementa un sencillo protocolo MAC basado en tiempos aleatorios de transmisión.

La tarea ReportTx es la que se ocupa de gestionar este módulo, haciendo uso del valor de lectura actualizado por la tarea Pulses, que monitoriza el sensor reed switch KSK-1A87-1015.

## 4 **TESTING**

Como ya se dijo en el apartado 1.2, uno de los objetivos particulares de este TFM consiste en el completo testeo del firmware de los microcontroladores (MCU) AT91SAM3SD8C [35] del Local Data Concentrator (LDC) y ATmega328P [36] del Endpoint (EP), tanto del programado directamente por el autor como del programado por otros compañeros del equipo Atmel. En relación con el segundo punto, la dificultad añadida que conlleva la revisión de desarrollos ajenos, se vio parcialmente aliviada por el hecho de que el autor también se encargó de la supervisión general del diseño firmware, lo que supuso un conocimiento detallado del mismo a lo largo de todo el proyecto.

Las pruebas se realizaron mediante rigurosos procedimientos de test, considerando los casos más improbables de error y tantas situaciones de coincidencia de eventos de diversa índole como pudieron ser concebidas. Algunos de estos procedimientos se plantearon como tests de larga duración, con objeto de probar el sistema en modo normal, mientras que otros se idearon para su ejecución en cortos periodos, ya que chequeaban funcionalidades concretas.

En cuanto se dispuso del hardware correspondiente y conforme se iba llevando a cabo la implementación de los nuevos protocolos de comunicación y la programación del firmware, se fueron ejecutando los protocolos de test diseñados, que se adecuaban al nivel de desarrollo alcanzado en cada fase.

Tras la ejecución de cada test se creaba una carpeta propia del test, incluyendo en ella el procedimiento completo aplicado al test y los resultados detallados con los comentarios oportunos. Además, se adjuntaba un plan de acciones para solventar los errores detectados y se anotaban las posibles mejoras que el test había sacado a la luz.

En el ANEXO C, apartado C.2, se recogen las distintas herramientas de test utilizadas.



## 5 **CONCLUSIONES**

La conclusión principal a la que se llega tras finalizar un proyecto de esta envergadura, integrado en el seno de un consorcio en el que rigen criterios de interoperabilidad, es la crucial importancia de una fluida y precisa comunicación entre las distintas partes, tanto a nivel externo como interno. Internamente, desde el punto de vista de la supervisión realizada por el autor de este TFM, la correcta definición de tareas, la colaboración entre los miembros del equipo y la exigencia de calidad en la ejecución del trabajo, han constituido los pilares del éxito.

Otra interesante conclusión, sujeta generalmente proyectos de este tipo, es que todo queda supeditado a los plazos exigidos por la empresa cliente y los recursos que la empresa encargada de llevarlo a cabo está dispuesta a dedicar.

Esto queda de manifiesto en la estructuración del firmware del LDC, donde se tuvo que adaptar una primera estructura que seguía la técnica Foreground/Background dentro de un RTOS (FreeRTOS), debido a los requisitos del planning. Lógicamente, lo ideal hubiera sido rehacer la estructura completa del FW según el criterio de diseño marcado por los RTOS.

Aunque en este trabajo se ha decidido no profundizar demasiado en el código, también cabe decir que con más tiempo las funciones y algoritmos programados podrían haber alcanzado un mayor grado de optimización, ahorrando recursos hardware, a pesar de que ya se ha tenido cuidado de trabajar en ese sentido.

En cuanto al testing, señalar que si la realización de test exhaustivos de los diseños desarrollados por uno mismo ya requiere de gran meticulosidad, el hecho de acometer esta tarea sobre programaciones ajenas entraña una considerable dificultad añadida, como se ha podido comprobar en este TFM.

Finalmente, cabe señalar que el entorno de trabajo que proporciona una empresa multinacional especializada en I+D+i, como es Atmel, resulta de gran ayuda en el desarrollo de proyectos como el que nos ocupa, aportando experiencia y soporte siempre que se necesitan.

### 5.1 **Consecución de Objetivos**

Tras llevar a cabo el completo desarrollo del sistema y las pruebas de testeo oportunas, se ha comprobado que se han cumplido en su totalidad los requisitos exigidos por el consorcio, teniendo actualmente la compañía energética en sus instalaciones la red piloto de telegestión de gas deseada, incluyendo los equipos desarrollados en este proyecto.

La definición de protocolos de comunicación tanto WM-Bus como DLMS/COSEM, ha resultado plenamente satisfactoria, constituyendo la base del nuevo estándar perseguido por la Comisión Nacional de Energía, con la única salvedad de algunos puntos de falta de interoperabilidad en la parte DLMS/COSEM, debido a pequeñas diferencias en tipos y formatos de programación, fácilmente corregibles.

Por último, las funcionalidades soportadas por los equipos dotan al sistema de AMM de gas de todas las características especificadas por el consorcio.

## **5.2 Líneas de Trabajo Futuras**

A continuación se desglosan las diferentes líneas para futuros desarrollos dentro del marco del proyecto:

- Completar la interoperabilidad en los nuevos protocolos DLMS/COSEM.
- Mejora del protocolo MAC para las comunicaciones RF implementado en los microcontroladores.
- Configuración y gestión del LDC directamente desde un HHT por RF mediante WM-Bus.
- Telecarga tanto del EP como del LDC por radio (WM-Bus) usando un HHT.
- Telecarga del EP desde los SSI, pasando por el GCT y a través del LDC.
- Empleo de técnicas de recopilación de energía (energy harvesting) en el EP.
- Desarrollo de un sistema similar con tecnología centrada en la frecuencia 169 MHz.
- Analizar los resultados y conclusiones entregados por la compañía energética de estos prototipos, tras probarlos en campo, para satisfacer las nuevas necesidades que les puedan surgir.

## 6 REFERENCIAS

- [1] <http://www.atmel.com/>. Página principal de Atmel Corporation.
- [2] <http://www.cne.es/cne/Home>. Página principal de la Comisión Nacional de Energía.
- [3] <http://www.gasnaturalfenosa.es/es/1285340760529/inicio.html>. Página principal de Gas Natural Fenosa.
- [4] <http://www.iberdrola.es/>. Página principal de Iberdrola.
- [5] <http://www.endesa.com/es/conoceendesa/nuestraestrategia/GrupoEnel>. Página principal de Enel Endesa.
- [6] <http://www.gazprom.com/about/subsidiaries/list-items/marketing-and-trading/>. Página de la filial Gazprom Marketing & Trading Ltd. del grupo Gazprom.
- [7] <http://www.britishgas.co.uk/>. Página principal de British Gas.
- [8] <http://www.eon.com/de.html>. Página principal de E.ON.
- [9] <http://www.grdf.fr/fr/>. Página principal de GrDF.
- [10] <http://www.erdfdistribution.fr/Accueil>. Página principal de ERDF.
- [11] <http://www.lyonnaise-des-eaux.fr/>. Página principal de Lyonnaise des Eaux.
- [12] <http://www.enemalta.com.mt/>. Página principal de Enemalta.
- [13] <http://www.prime-alliance.org/>. Página principal de PRIME Alliance.
- [14] A. Sendin, I. Berganza, A. Arzuaga, A. Pulkkinen, I. H. Kim. "Performance Results from 100,000+ PRIME Smart Meters Deployment in Spain". 2012.
- [15] <http://www.metersandmore.eu/index.asp>. Página principal de METERS AND MORE Association.
- [16] <http://www.nuritelecom.com/about/successes-Gas-Natural.php>. Proyecto AMM de gas de Nuri Telecom para Gas Natural Fenosa. 2009.
- [17] <http://www.erdfdistribution.fr/recherche/?q=linky>. Proyecto Linky de ERDF.
- [18] <http://www.enemalta.com.mt/index.aspx?cat=3&art=218>. Gas Project de Enemalta.
- [19] <http://www.enemalta.com.mt/index.aspx?cat=20&art=125>. Proyecto Integrated Utilities Business Systems (IUBS) de Enemalta.
- [20] K. H. Chang, B. Mason. "The IEEE 802.15.4g Standard for Smart Metering Utility Networks". 2012.
- [21] <http://www.opennode.eu/>. Página principal de OpenNode.
- [22] <http://www.openmeter.com/>. Página principal de OPEN Meter.
- [23] <http://www.addressfp7.org/>. Página principal de ADDRESS.
- [24] <http://www.ietf.org/>. Página principal de IETF.
- [25] <http://www.zigbee.org/>. Página principal de ZigBee Alliance.
- [26] <http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>. Página principal de Bluetooth.
- [27] <http://www.wi-fi.org/>. Página principal de Wi-Fi Alliance.
- [28] UNE-EN\_13757. M-Bus.
- [29] Open Metering System Specification Vol.1, 2 and 3. 2011.
- [30] Atmel\_WM-Bus\_Messages&Sequences (confidencial).
- [31] Dutch Smart Meter Requirements.
- [32] <http://www.dlms.com/index2.php>. Página principal de DLMS User Association.
- [33] Companion Specification for Energy Metering (COSEM), Blue, Green, Yellow and White books (confidencial).
- [34] DataModel\_Companion\_v3 (confidencial).
- [35] <http://www.atmel.com/Images/doc11090.pdf>. Hoja de características o Data Sheet del microcontrolador AT91SAM3SD8C.

- [36] [http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet.pdf). Hoja de características o Data Sheet del microcontrolador ATmega328P.
- [37] J. D. Muñoz Frías. "Sistemas\_Empotrados\_en\_Tiempo\_Real". 2009.
- [38] <http://www.freertos.org>. Página principal de FreeRTOS.
- [39] InterfaceUSI\_User\_Manual.
- [40] Atmel\_Backdoor\_Documentation (confidencial).



## 7 GLOSARIO

Para simplificar la lectura del texto, el documento referencia las distintas tecnologías, interfaces, técnicas, protocolos, instituciones, etc. mediante acrónimos comúnmente aceptados.

A continuación se muestra una lista de acrónimos con su correspondiente significado:

- **ADC:** Analog Digital Converter.
- **ADDRESS:** Active Distribution networks with full integration of Demand and distributed energy RESourceS.
- **AES:** Advanced Encryption Standard.
- **AMI:** Advanced Metering Infrastructure.
- **AMM:** Automatic Meter Management.
- **AMR:** Automatic Meter Reading.
- **API:** Application Programming Interface.
- **ASCII:** American Standard Code for Information Interchange.
- **CCT:** Concentrador de Centro de Transformación.
- **CEN:** Comité Européen de Normalisation.
- **CENELEC:** Comité Européen de Normalisation Electrotechnique.
- **CNAF:** Cuadro Nacional de Atribución de Frecuencias.
- **COSEM:** Companion Specification for Energy Metering.
- **DLMS:** Device Language Message Specification.
- **DR:** Daily Report.
- **DSMR:** Dutch Smart Meter Requirements.
- **EOL:** End Of Line.
- **EP:** Endpoint.
- **FTL:** Flash Translation Layer.
- **FU:** Firmware Upgrade.
- **FW:** Firmware.
- **GCT:** Gestor del Centro de Transformación.
- **GNF:** Gas Natural Fenosa.
- **GW:** Gateway.
- **HDLC:** High-Level Data Link Control.
- **HHT:** Hand Held Terminal.
- **HW:** Hardware.
- **I+D+i:** Investigación, Desarrollo e innovación.
- **ID:** Identifier.
- **IDE:** Integrated Development Environment.
- **IETF:** Internet Engineering Task Force.
- **IO:** Input Output.
- **IP:** Internet Protocol.
- **ISR:** Interrupt Service Routine.
- **LAN:** Local Area Network.
- **LDC:** Local Data Concentrator.
- **MAC:** Medium Access Control.
- **MAN:** Metropolitan Area Network.
- **M-Bus:** Meter Bus.

- **MCU:** Micro Controller Unit.
- **MFT:** Atmel Metering Field Tool.
- **MI:** Meter Interface.
- **MUC:** Multi Utility Communication Controller.
- **NRTOD:** Near Real Time On Demand.
- **NVIC:** Nested Vectored Interrupt Controller.
- **OMS:** Open Metering System.
- **OPEN:** Open Public Extended Network.
- **PC:** Personal Computer.
- **PLC:** Power Line Communications.
- **PRIME:** PoweRline Intelligent Metering Evolution.
- **RF:** Radio Frequency o Radiofrecuencia.
- **RFI:** Radio Frequency Interface.
- **RSSI:** Receive Signal Strength Indication.
- **RTOD:** Real Time On Demand.
- **SSI:** Sistemas de Información.
- **SUN:** Smart Utility Networks.
- **SW:** Software.
- **TC:** Timer Counter.
- **TCP:** Transmission Control Protocol.
- **TFM:** Trabajo Fin de Máster.
- **TWI:** 2-wire Serial Interface.
- **UART:** Universal Asynchronous Receiver Transmitter.
- **USART:** Universal Synchronous Asynchronous Receiver Transmitter.
- **USB:** Universal Serial Bus.
- **USI:** Universal Serial Interface.
- **WAN:** Wide Area Network.
- **WM-Bus:** Wireless M-BUS.

## **ANEXO A PLANIFICACIÓN**

### **A.1 Fases y tareas**

En el siguiente esquema se detallan las fases, tareas e hitos del proyecto, pero acotadas a las tratadas en este trabajo:

- F0.- Vigilancia tecnológica (compañía energética).
  - T0.1.- Realización sistemática de la captura, el análisis, la difusión y la explotación de informaciones científicas y técnicas de interés.
  - T0.2.- Aviso sobre las innovaciones científicas o técnicas susceptibles de crear oportunidades o amenazas.
- F1.- Autorización del proyecto I+D+i.
  - T1.1.- Designación de responsables.
- F2.- Diseño básico.
  - T2.1.- Definición de objetivos.
  - T2.2.- Análisis de requisitos (especificaciones de concurso).
  - T2.3.- Perfilado de características de diseño.
  - T2.4.- Realización de esquemas y planos preliminares.
- H1
- F3.- Planificación.
  - T3.1.- Asignación de recursos materiales y humanos.
  - T3.2.- Asignación de infraestructura.
  - T3.3.- Estructura de comunicaciones.
  - T3.4.- Definición de fases y tareas.
- H2
- F4.- Diseño detallado.
  - T4.1.- Definición del nuevo protocolo RF.
    - ST4.1.1) Estudio de protocolos base.
    - ST4.1.2) Desarrollo de nuevo protocolo.
  - T4.2.- Definición de los nuevos objetos DLMS/COSEM.
    - ST4.2.1) Estudio de protocolos base.
    - ST4.2.2) Desarrollo de nuevo protocolo.
  - T4.3.- Programación del firmware del LDC.
    - ST4.3.1) Análisis de módulos necesarios.
    - ST4.3.2) Diseño de estructura.
    - ST4.3.3) Programación.
    - ST4.3.4) Versión provisional.
    - ST4.3.5) Versión definitiva.
  - T4.4.- Programación del firmware del EP.
    - ST4.4.1) Análisis de módulos necesarios.
    - ST4.4.2) Diseño de estructura.
    - ST4.4.3) Programación.
    - ST4.4.4) Versión provisional.
    - ST4.4.5) Versión definitiva.
  - T4.5.- Redacción de la documentación y memoria del TFM.
- H3
- F5.- Pruebas piloto.
  - T5.1.- Ensayos y pruebas internos.
- H4
- F6.- Demostración.
  - T6.1.- Compañía energética.
    - ST6.1.1) Presentación y entrega de los equipos.

- ST6.1.2) Presentación y entrega de la documentación de proyecto.
- T6.2.- Universidad de Zaragoza.
  - ST6.2.1) Presentación de los equipos.
  - ST6.2.2) Presentación de la memoria del TFM.
- H5
- F7.- Cierre.
  - T7.1.- Conclusión del proyecto.
  - T7.2.- Lecciones aprendidas.
- H6

Nota: *F: Fase; T: Tarea; ST: SubTarea; H: Hito.*

## **A.2 Cronograma**

En la Figura 6 se presenta un cronograma, a modo de diagrama de Gantt, que muestra el ciclo de vida de las fases y tareas del proyecto propias de este trabajo. Realmente el cronograma empieza en la fase 4, aunque también se incluyan las fases anteriores para aportar una mayor claridad, ya que el TFM que nos ocupa abarca de ahí en adelante.

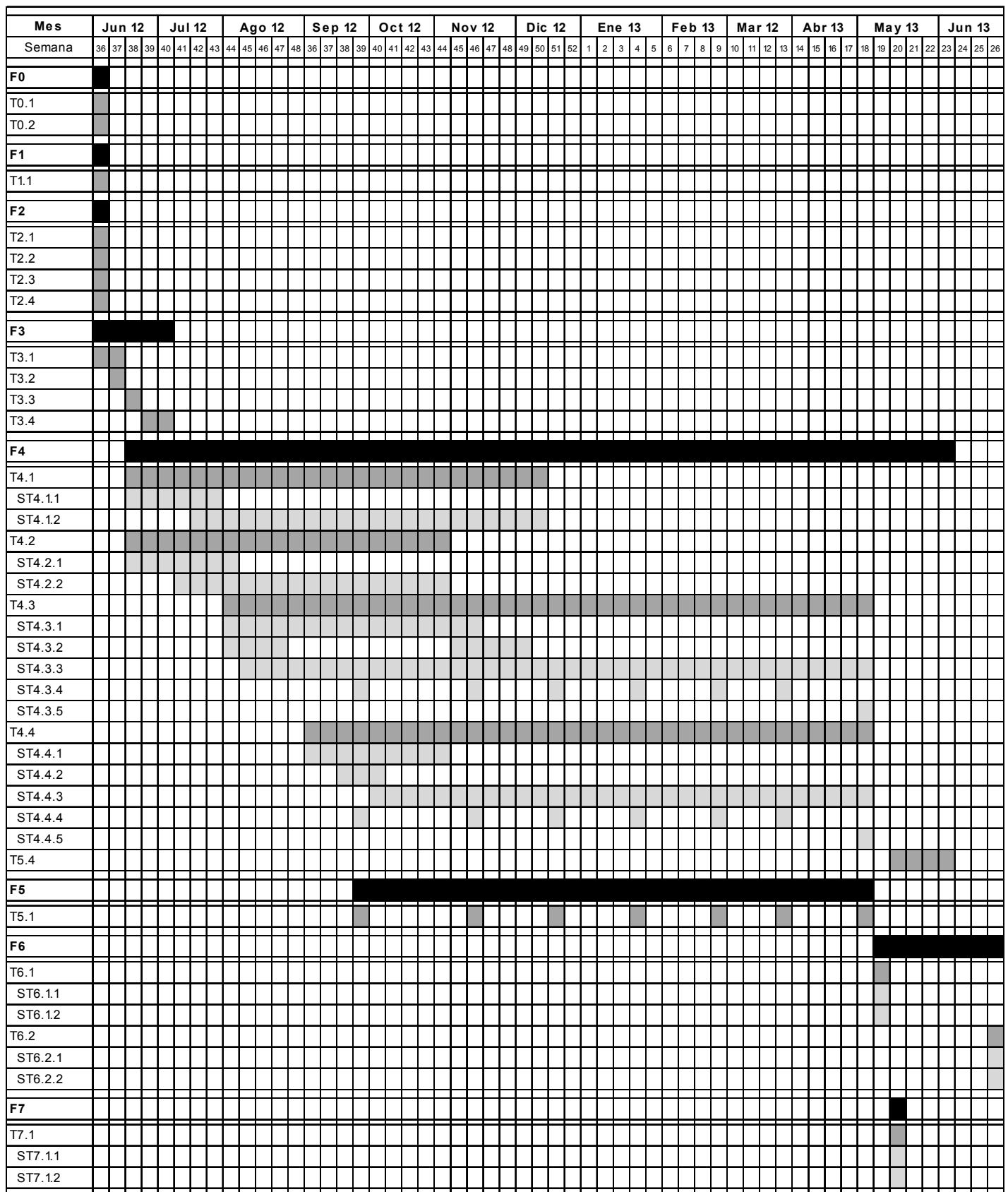


Figura 6. Cronograma.



## ANEXO B HARDWARE

Por asentar las bases sobre las que se debe desarrollar el firmware de los equipos, se presentan en este apartado los esquemas hardware (Figura 7 y Figura 8) junto con unas breves explicaciones.

En ambos casos se incluyen LED de usuario, ligados a puertos IO, cuyas funciones no se detallan en este documento por restricción de la compañía energética.

Información detallada sobre el hardware se recoge en el TFM centrado en esa parte.

### B.1 Local Data Concentrator

#### B.1.1 Esquema de placa

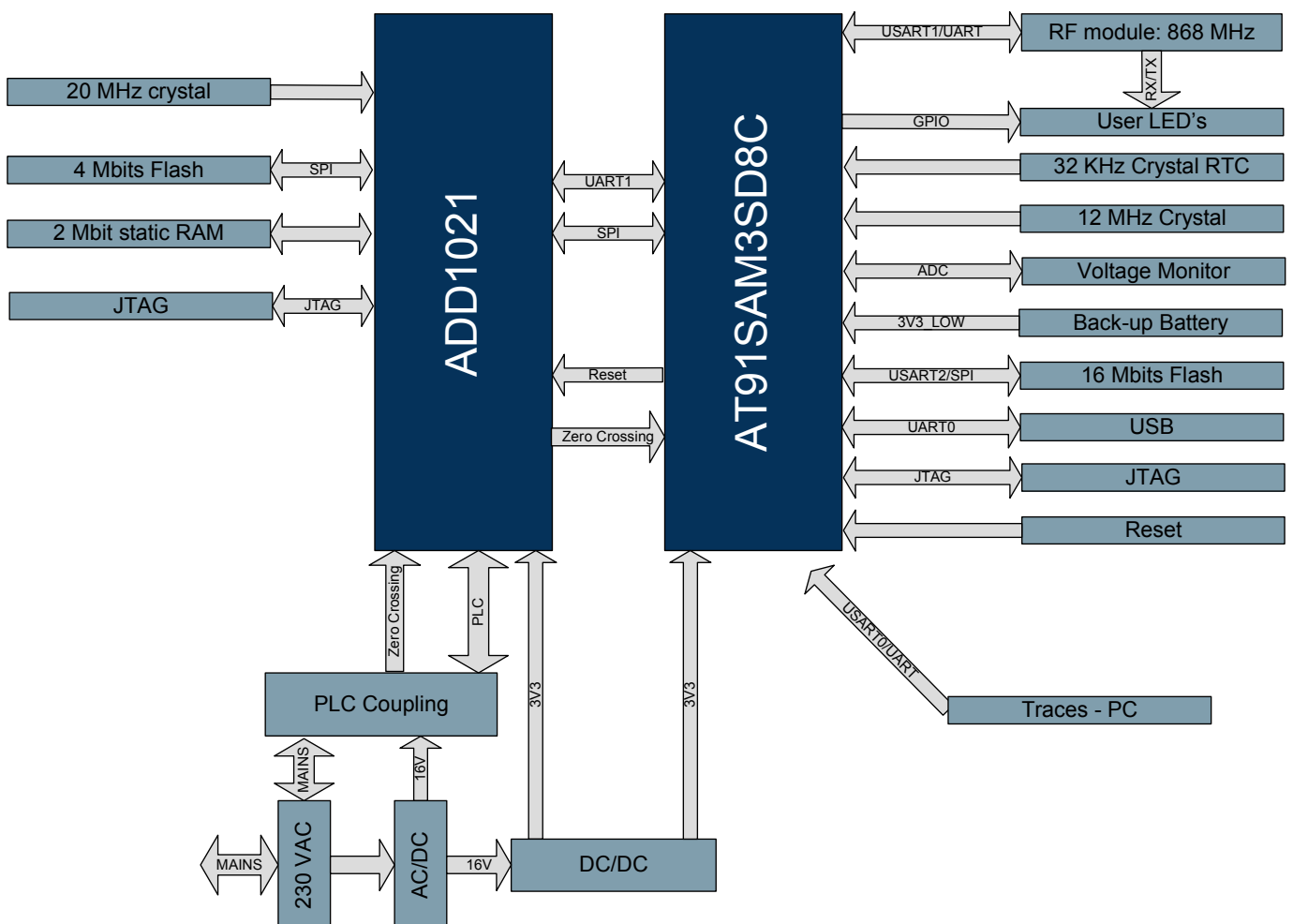


Figura 7. Esquema HW del LDC.

La programación empotrada objeto de este trabajo se ha realizado en el microcontrolador (MCU) AT91SAM3SD8C de 32 bits, aunque se ha usado el MCU ADD1021 ya programado en Atmel, para soportar las comunicaciones PLC y PRIME.

### B.1.2 Compatibilidad

El hardware del LDC para el que se ha llevado a cabo la programación embebida es la versión ADD1021-RF v3.

## B.2 Endpoint

### B.2.1 Esquema de placa

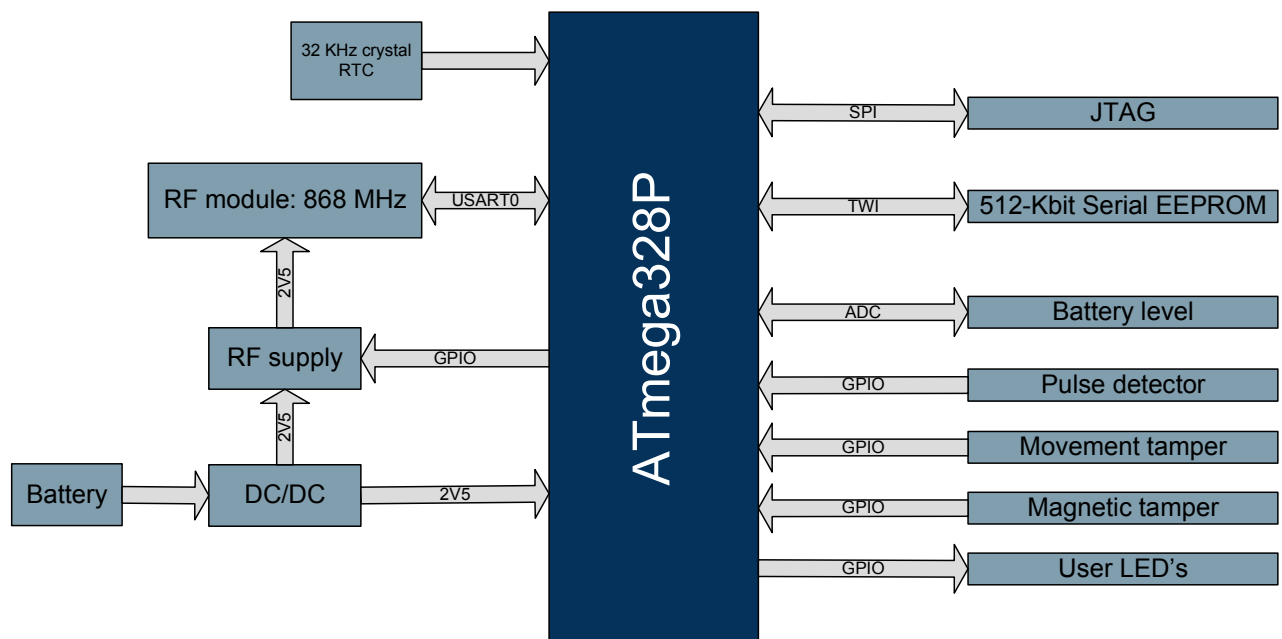


Figura 8. Esquema HW del EP.

En este caso sólo hay un MCU, el ATmega328P de 8 bits, en el que se implementan todas las funcionalidades requeridas.

### B.2.2 Compatibilidad

El hardware del EP para el que se ha llevado a cabo la programación embebida es la versión ATPLRFEP868-99 v4.



## ANEXO C HERRAMIENTAS DE DESARROLLO

En este anexo se recogen las principales herramientas de desarrollo utilizadas para la realización de las tareas incluidas en este TFM.

### C.1 Firmware

#### C.1.1 Local Data Concentrator

##### C.1.1.1 IAR

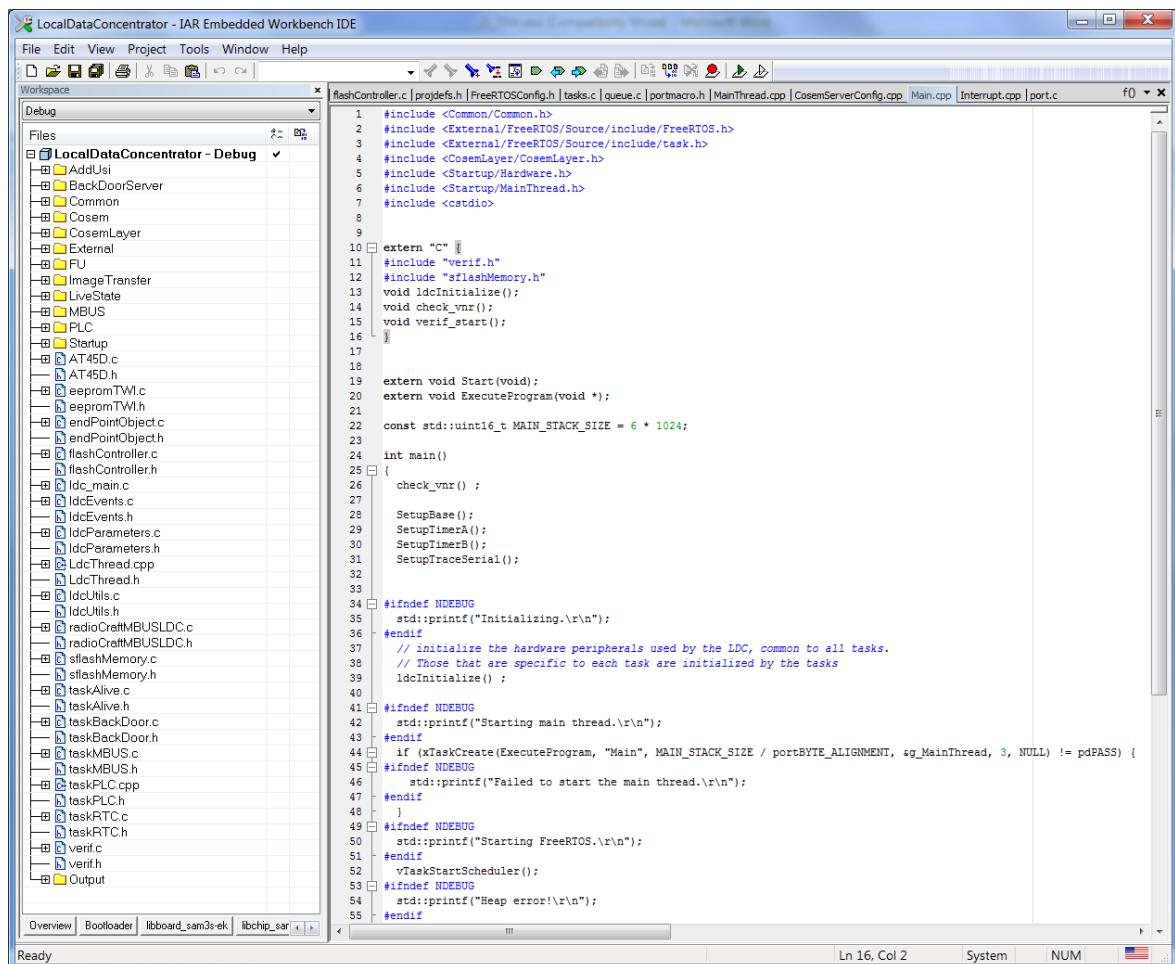


Figura 9. IDE de IAR Embedded Workbench.

La principal herramienta empleada para la programación del firmware es IAR Embedded Workbench para ARM. Su entorno de desarrollo integrado (IDE), presentado en la Figura 9, incluye:

- Editor and project manager.
- Compilador C/C++ y librerías.

- Debugger: tanto sobre objetivo como en simulación.

La versión de IAR utilizada es la 6.50 y el microcontrolador objeto de la programación empotrada es el ARM Cortex - M3 de 32 bits AT91SAM3SD8C.

El IDE es un completo entorno de programación basado en workspaces capaces de albergar varios proyectos con sus correspondientes ficheros, editor de texto, mensajes de compilación, puntos de ruptura o breakpoints, distintos niveles de búsqueda, etc.

### C.1.1.2 Programador J-Link

Para realizar las descargas de firmware en el micro, así como las pruebas en modo depuración o debug, se ha hecho uso de un emulador USB JTAG de Segger Microcontroller System llamado J-Link (Figura 10). Este programador conecta el PC donde corre IAR con la placa del LDC, permitiendo las tareas de descarga del código y su monitorización en ejecución desde el entorno IAR.



Figura 10. Programador J-Link.

```

J-Link Commander
SEGGER J-Link Commander V4.51k ('?' for help)
Compiled Aug 1 2012 15:24:28
DLL version V4.51k, compiled Aug 1 2012 15:24:11
Firmware: J-Link ARM V7 compiled Jun 20 2012 19:45:53
Hardware: V7.00
S/N: 157000503
OEM: IAR
VTarget = 0.000V
J-Link>r
Info: TotalIRLen = 4, IRPrint = 0x01
Info: Found Cortex-M3 r2p0, Little endian.
Info: TPIU fitted.
Info: FPUnit: 6 code (BP) slots and 2 literal slots
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Info: Found Cortex-M3 r2p0, Little endian.
Info: TPIU fitted.
Info: FPUnit: 6 code (BP) slots and 2 literal slots
J-Link>exec device = AT91SAM3SD8C
Info: Device "AT91SAM3SD8C" selected (512 KB flash, 64 KB RAM).
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
J-Link>h
R0 = 400E1400, R1 = 00000000, R2 = A5000004, R3 = 004110E1
R4 = 200011A8, R5 = 00000210, R6 = 200012B9, R7 = 00000210
R8 = 00000111, R9 = 00000210, R10 = 00000000, R11 = 200011A8
R12 = 00000004, R13 = 20000200, MSP = 20000200, PSP = 20001168
R14(LR) = FFFFFFFF, R15(PC) = 00401200
XPSR = 01000000, APSR = 00000000, EPSR = 01000000, IPSR = 00000000
CFBP = 00000000, CONTROL = 00, FAULTMASK = 00, BASEPRI = 00, PRIMASK = 00
CycleCnt = 0381F866
J-Link>loadbin c:\data\ldc.bin, 0x00402000

```

Figura 11. Entorno JLink-Commander.

El J-Link utiliza el puerto serie JTAG Debug definido en el MCU para estos cometidos.

### C.1.1.3 J-Link Commander

Segger ofrece un paquete firmware para el programador J-Link. Entre las funcionalidades incluidas en este paquete resulta de gran interés el JLink-Commander (Figura 11), una consola de línea comandos que permite el acceso directo al microcontrolador posibilitando descargar el firmware fácil y rápidamente. El uso de este tipo de aplicaciones es muy común en finales de línea de procesos productivos o End Of Line (EOL).

## C.1.2 Endpoint

### C.1.2.1 IAR

Al igual que con el LDC, la principal herramienta empleada ha sido IAR Embedded Workbench, pero en este caso para AVR. Las características de su entorno de desarrollo integrado (IDE) son similares a las ya expuestas en el apartado C.1.1.1:

La versión de IAR utilizada es la 6.20 y el microcontrolador objeto de la programación empotrada es el AVR RISC de 8 bits ATmega328P.

### C.1.2.2 Programador JTAGICE mkII

Para realizar las descargas de firmware en el micro, así como las pruebas en modo depuración o debug, se ha hecho uso de un emulador USB JTAG de Atmel Corporation llamado JTAGICE mkII (Figura 12). Este programador conecta el PC donde corre IAR con la placa del EP, permitiendo las tareas de descarga del código y su monitorización en ejecución desde el entorno IAR.

El JTAGICE mkII utiliza el puerto SPI del MCU para realizar sus funciones.



Figura 12. Programador JTAGICE mkII.

### C.1.2.3 Atmel Studio

Atmel Studio es la plataforma de desarrollo integrada de Atmel Corporation, que permite el desarrollo y depuración de aplicaciones para los microcontroladores de Atmel ARM Cortex - M y

AVR. Su IDE tiene similares características a las de IAR (ver apartado C.1.1.1). Se ha utilizado la versión Atmel Studio 6 (Figura 13).

La funcionalidad empleada en este proyecto ha sido la carga directa de código en el MCU desde archivos en binario, previamente generados mediante IAR.

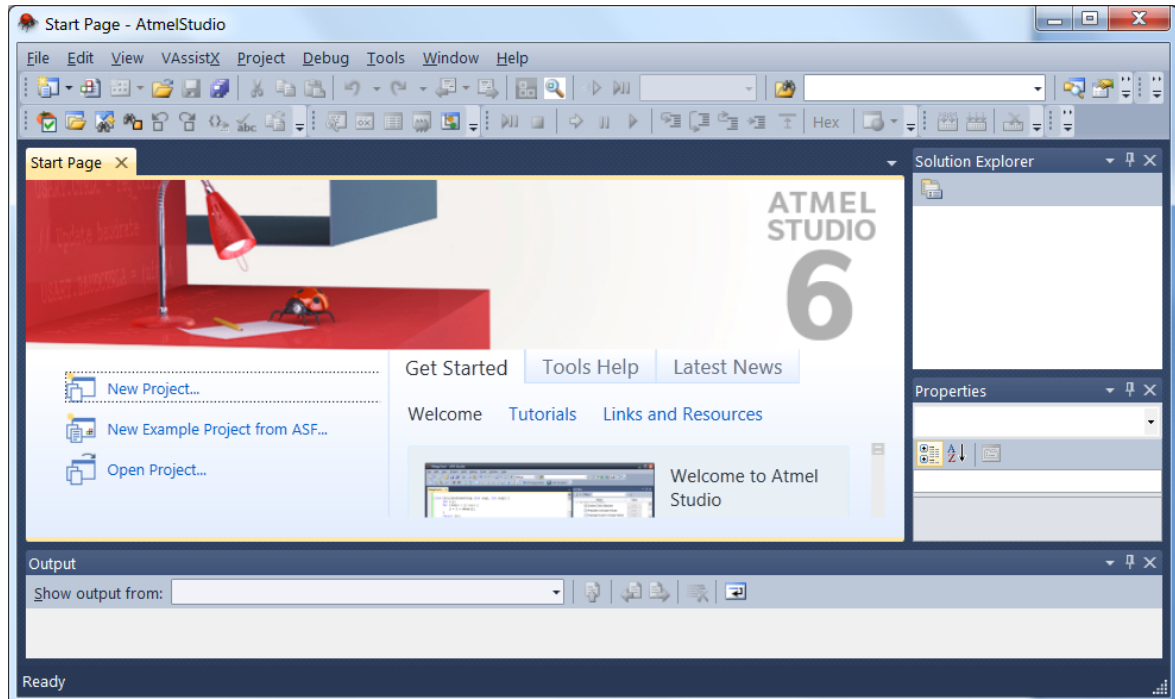


Figura 13. Atmel Studio 6.

## C.2 Configuración y pruebas

### C.2.1 Hand Held Terminal

El Hand Held Terminal (HHT) es un equipo portátil para las tareas de configuración y gestión, tanto del EP por como del LDC/GW. Está formado por un dispositivo RF llamado Radio Frequency Interface (RFI), conectado a un PC por USB para ser gestionado por una aplicación llamada Metering Field Tool (MFT). La comunicación con los EP es por RF y con los LDC/GW por puerto serie.

#### C.2.1.1 RFI

Equipo de configuración y gestión de los EP. Dado que los EP sólo se comunican por RF mediante WM-Bus, el RFI incorpora este tipo de comunicación con los nuevos protocolos creados en este trabajo. Requiere de la MFT, comentada en el apartado C.2.1.2, a la que se conecta por USB. En la Figura 14 se muestra una foto del RFI.

*Nota: Aunque el RFI ha sido objeto de desarrollo en este proyecto, se incluye en este capítulo porque permite la comprobación de diversas funcionalidades.*



Figura 14. RFI.

### C.2.1.2 Metering Field Tool 4.0.0

La Metering Field Tool (MFT) es una aplicación PC de configuración y gestión de equipos en la red, cuyo interfaz de inicio se muestra en la Figura 15. Junto con el RFI constituye el equipo de campo HHT requerido en las especificaciones. Permite dos modos de funcionamiento:

- **Modo RFI:** la aplicación se conecta a un RFI mediante un puerto USB del PC. A través del RFI es posible realizar las tareas de configuración y gestión de los EP, a los que sólo es posible acceder por RF (WM-Bus).
- **Modo LDC:** la aplicación se conecta a un LDC mediante un puerto USB del PC. En este caso es posible realizar directamente las tareas de configuración y gestión del LDC, incluyendo las relacionadas con los EP instalados en él (no necesita RFI).

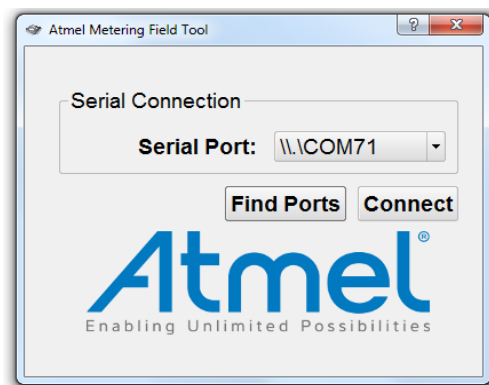


Figura 15. Interfaz de inicio de la MFT.

*Nota: Aunque la MFT ha sido objeto de desarrollo en este proyecto por otros miembros del equipo de Atmel, se incluye en este capítulo porque permite la comprobación de diversas funcionalidades.*

### C.2.2 Sniffer WM-Bus

Monitor de red wireless M-Bus (Sniffer) iM871A IMST (Figura 16), válido para la frecuencia 868 MHz, con el software correspondiente, Wireless M-Bus Studio 0.1.9 (Figura 17), para la captura de mensajes en ese protocolo y a esa frecuencia.

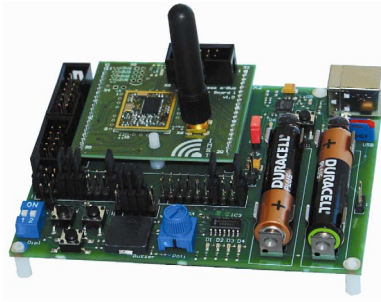


Figura 16. Sniffer WM-Bus iM871A IMST.

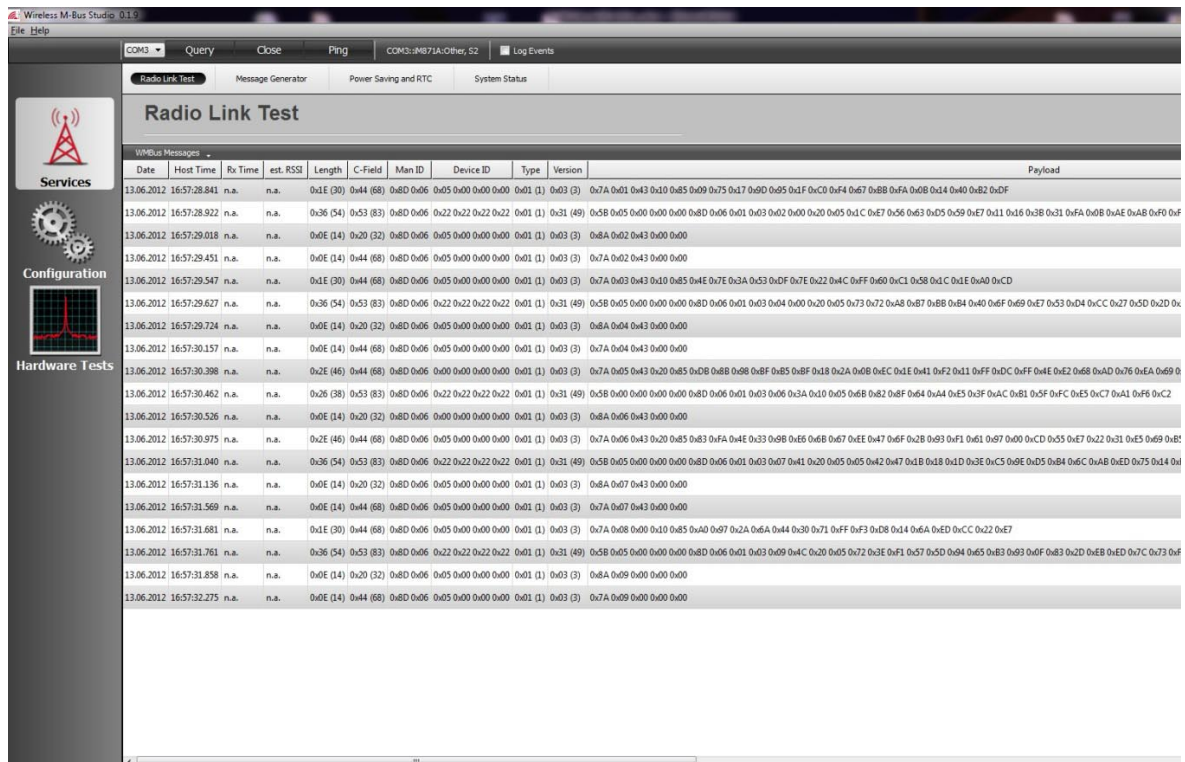


Figura 17. Wireless M-Bus Studio 0.1.9.

### C.2.3 Aplicaciones Atmel para PLC y PRIME

Se ha hecho uso de las siguientes aplicaciones Atmel relacionadas con PLC y PRIME:

- ADD6030v30 v1.6.09: para carga del firmware de PRIME en microcontroladores (Figura 18).
- PLC Multiprotocol Sniffer v1.3.0: sniffer PLC.
- PRIME Network Manager v1.0.3: monitor de red PLC PRIME.

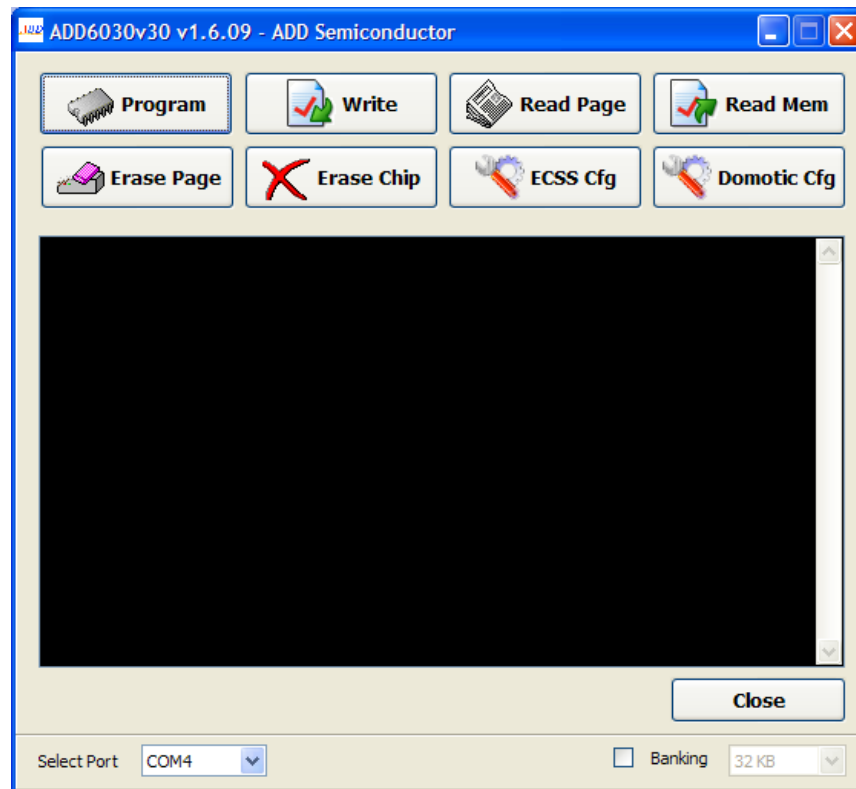


Figura 18. ADD6030v30 v1.6.09.

## C.2.4 Phenix

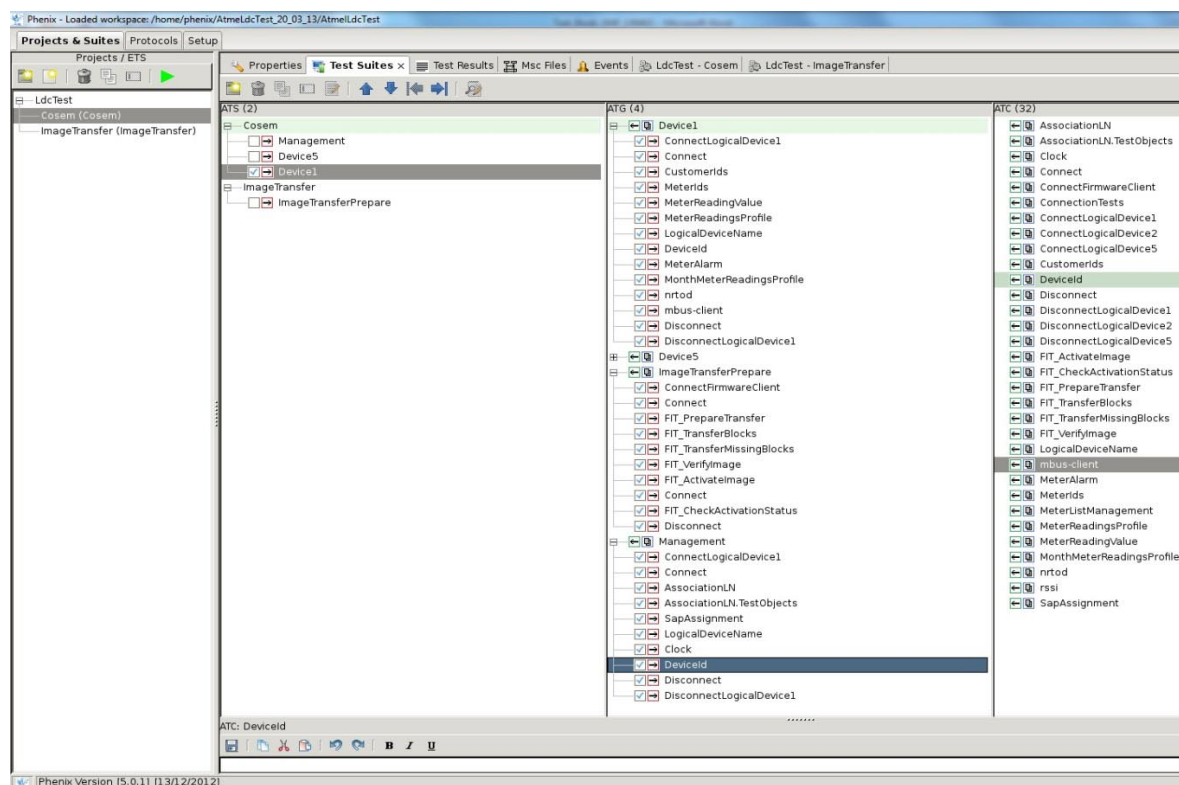


Figura 19. Vista de los test suites de Phenix.



Simulador DLMS/COSEM para emular las funcionalidades de los nuevos objetos de gas. Conectado al LDC por puerto serie mediante HDLC (no por PLC PRIME).

Phenix permite la simulación de las comunicaciones DLMS/COSEM con un LDC, haciendo el papel del GCT. Se basa en test suites, que consisten en tests completos de un objeto COSEM. Varios test suites pueden ser lanzados al mismo tiempo. Ejemplos de test suites se muestran en la Figura 19.

### C.2.5 Gestor de Centro de Transformación

Se ha utilizado el equipo CCT4 Concentrador de Centro de Transformación para sistema ARM para llevar a cabo pruebas reales de conexión a un GCT. La comunicación es por PLC PRIME y la gestión a través de un interfaz web (Figura 20).

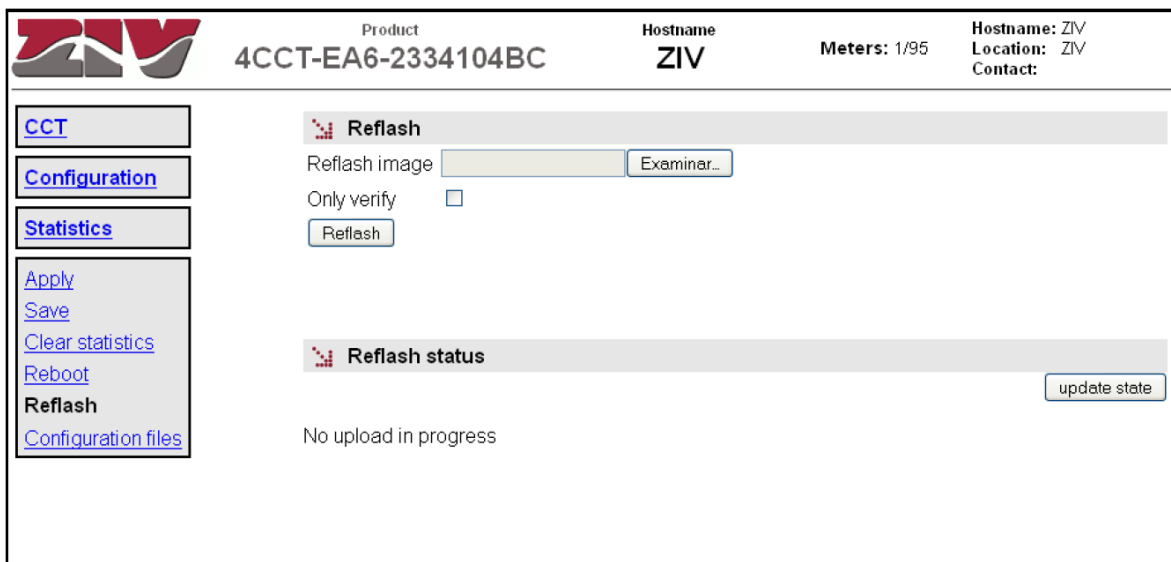


Figura 20. Interfaz web de CCT4.

### C.2.6 Imán

Se requiere de un imán para lanzar determinadas secuencias de comunicación en los EP.

### C.2.7 Compresor de aire

Utilizado para generar efectos de flujo rápido en los contadores de gas y probar así la alarma de High Flow del EP.

### C.2.8 Fuentes de alimentación

Aparte de para poder hacer pruebas sin consumir pilas, las fuentes de alimentación se han usado para simular la alarma de Low Power del EP, aunque ésta también se ha probado directamente con pilas.



### C.2.9 Contadores de gas

Los contadores de gas son necesarios para las pruebas de funcionamiento real. Siguiendo las especificaciones del concurso, se han empleado los dos tipos siguientes (Figura 21):

- Itron Actaris GallusG46m3/h7/8".
- Elster BK-G4V1,2A160G7/8"0,5.



Figura 21. Contadores de gas marcas Itron (izda.) y Elster (dcha.).

### C.2.10 Ordenadores personales

Lógicamente, para la ejecución de todas las aplicaciones mencionadas anteriormente han sido necesarios ordenadores personales o PC.